

Joona Melartin

VIDEOPELIN JATKO-OSAN SUUNNITTELU JA PROTOTYPOINTI

VIDEOPELIN JATKO-OSAN SUUNNITTELU JA PROTOTYPOINTI

Joona Melartin
Opinnäytetyö
Syksy 2016
Tietojenkäsittelyn koulutusohjelma
Oulun ammattikorkeakoulu

TIIVISTELMÄ

Oulun ammattikorkeakoulu

Tietojenkäsittelyn koulutusohjelma, web-sovellusten kehittäminen

Tekijä: Joona Melartin

Opinnäytetyön nimi: Videopelin jatko-osan suunnittelu ja prototypointi

Työn ohjaaja: Matti Viitala

Työn valmistumislukukausi ja -vuosi: Syksy 2016

Sivumäärä: 66

Tämä opinnäytetyö käsittelee pelisuunnittelua jatko-osan suunnittelun näkökulmasta. Työn tavoitteena on analysoida tekijän vanhempi teos vuodelta 2014 ja tehdä sen jatko-osalle pelisuunnitteludokumentti sekä pieni prototyyppi.

Opinnäytetyö jakautuu teoriaan, ensimmäisen pelin analyysiin, suunnitteluraporttiin, prototyypin raporttiin sekä lopputuloksiin. Teoriaosiossa käsitellään syitä ja lähtökohtia jatko-osan suunnittelulle samalla esittäen eroja ensimmäisen pelin ja jatko-osan suunnittelun välillä. Ensimmäisen pelin analyysi osio sisältää kuvauksen ensimmäisestä pelistä, takaisinmallinnustulokset sekä jatko-osan suunnittelun lähtökohtien määrytykset. Suunnitteluraportti osuus esittelee suunnitellun jatko-osan suunnitteluprosessia ja pelisuunnitteludokumentin sisällön yleisesti. Prototyypin raportti osio käy läpi prototyypin kehitysprosessia pelisuunnitteludokumentin pohjalta. Lopputulos osio käsittelee työn tuloksia jatkokehityksen kannalta.

Pelisuunnitteludokumentti ja pieni prototyyppi tehtiin onnistuneesti. Pelisuunnitteludokumentti päätyi kattamaan paljon yksityiskohtia kokonaisen pelin kehitystä varten. Prototyyppi suunniteltiin sisältämään vain perustavat mekaniikat pelistä ja päätyi täyttämään suunnitelmat onnistuneesti.

Vaikka pelisuunnitteludokumentin kirjoittaminen ja prototyypin kehitys olivat onnistuneita, prototyyppi jäi liian pieneksi, jotta voisi päätellä onko pelin idea täysin toimiva ja hauska. Tästä huolimatta jatkokehitys olisi helppoa, sillä sekä pelisuunnitteludokumentti ja prototyyppi antavat sille hyvän pohjan.

Asiasanat: peliohjelmointi, pelianalyysi, pelit, prototyypit

ABSTRACT

Oulu University of Applied Sciences
Business Information Systems, degree option of Web Application Development

Author: Joona Melartin

Title of thesis: Designing video-game sequel and development of its prototype

Supervisor: Matti Viitala

Term and year when the thesis was submitted: Autumn 2016 Number of pages: 66

This Bachelor's thesis looks at game design from the perspective of designing a sequel to a video game. The objective of the thesis was to analyze the author's older creation from 2014 as well as to create a game design document and a small prototype for its sequel.

The thesis consists of theory, the first game analysis, a design report, a prototype report and results sections. The theory section discusses reasons and starting points for designing a sequel along with presentation of the differences between designing a first game and a sequel. The first game analysis section contains a description of the first game, reverse engineering results and a definition of the starting points for designing a sequel. The design section presents the designing process of the planned sequel and the overall contents of the game design document. The prototype section goes through the process of creating the prototype based on the game design document. The results section discusses the results of the thesis from the perspective of further development.

The game design document and a small prototype were completed successfully. The game design document ends up covering a lot of details needed for development of a full game. The prototype was planned to contain only the basic mechanics of the game and it successfully implemented the plans.

While writing the game design document and development of the prototype were successful, the prototype was too small in order to determine if the idea of the game is fully functional and fun. However, further development would be easy as both the game design document and the prototype offer good basis for that.

Keywords: game programming, game analysis, games, prototyping

SISÄLLYS

1	JOHDANTO	7
2	JATKO-OSAN LÄHTÖKOHDAT	8
2.1	Miksi jatko-osa?	8
2.2	Ero uuden pelin ja jatko-osan suunnittelun välillä	9
2.3	Suunnittelun lähtökohdat	10
2.3.1	Palaute, kohderyhmä ja kilpailu	10
2.3.2	Samankaltaisuus ja ideoiden kierrätys	12
2.3.3	Yhteenveto	13
3	ENSIMMÄINEN PELI	15
3.1	Pelin yleiskuvaus	15
3.2	Takaisinmallinnus	16
3.2.1	Unity 3D Obfuscator	18
3.2.2	ILSpy	20
3.2.3	Unity Asset Extractor	22
3.2.4	Päätelmät	23
3.3	Palaute	24
3.4	Jatko-osan lähtökohdat	24
3.4.1	Palaute ja ehdotukset	25
3.4.2	Kohderyhmä	26
3.4.3	Kilpailu	27
3.4.4	Yhteenveto	29
4	JATKO-OSAN SUUNNITTELU	30
4.1	Pelisuunnitteludokumentti ja sen rakenne	30
4.2	Aivoriihi	32
4.3	Pelin yleiset asiat	34
4.4	Pelattavuus	37
4.5	Käyttöliittymä	42
4.6	Grafiikka ja ääni	43
4.7	Kenttäsuunnittelu	45
4.8	Lopuksi	47
5	JATKO-OSAN PROTOTYYPIN KEHITYS	48

5.1	Rajaus ja valmistelut	48
5.2	Tuotanto	51
6	TYÖN LOPPUTULOS.....	63
	LÄHTEET	65

1 JOHDANTO

Tässä opinnäytteessä on tarkoituksena käydä läpi pelisuunnittelua jatko-osan näkökulmasta. Työn aikana tehtiin yksi suunnittelupainotteinen toteutus kirjoittajan yhdestä vanhemmasta teoksesta. Tämän lisäksi suunnittelun pohjalta tehtiin prototyyppi, jonka kautta tarkasteltiin suunnitelmia uudelleen jatkokehityksen kannalta.

Aihe valittiin kiinnostuksen ja intohimon pohjalta; työ olisi liittynyt videopelien tekoon joka tapauksessa. Jatko-osien suunnittelu, motivaatiot ja teko ovat hiukan tuntemattomampi aihe ja täten kiinnostus työhön oli korkea. Työllä ei ole työelämään muuta yhteyttä, kuin mahdollinen portfoliomateriaali. Työ tehtiin itsenäisen pelikehittäjän sekä tietojenkäsittelyopiskelijan näkökulmasta.

Työ jakautuu viiteen osaan: jatko-osan suunnittelun teoriaan, ensimmäisen pelin kuvaukseen, jatko-osan suunnitteluun, prototyyppiin ja arviointiin. Teoriaosuudessa käydään läpi syitä ja lähtökohtia pelisuunnittelussa, kun ollaan tekemässä jatko-osaa videopelille. Ensimmäisen pelin kuvauksessa käydään läpi ensimmäisen pelin olemusta, rakennetta, sen saamaa palautetta sekä lähtökohtia jatko-osan kehityksen kannalta. Suunnittelu-osuudessa aiemman teorian ja pelianalyysin pohjalta kirjoitetaan pelisuunnitteludokumentti. Prototyyppi-osuudessa taas tehdään aiemmin kirjoitetun pelisuunnitteludokumentin pohjalta prototyyppi, jota taas arviointi-osuudessa tarkastellaan jatkokehityksen kannalta.

Tämä raportti koostuu työvaiheiden raportoinnista, mutta teoriaosuuksissa käytetty tietoperusta koostuu Internetistä löydetystä lähteistä. Samoja lähteitä käytetään myös perustelemaan työvaiheissa tehtyjä päätöksiä. Osa päätöksistä, perusteluista sekä teoriasta pohjautuu kuitenkin kirjoittajan omiin ajatuksiin, kokemuksiin ja päätelmiin. Kirjoittajan ajatukset ja kokemukset perustuvat useamman vuoden harrastustoimintaan pelisuunnittelun ja pelikehityksen parissa. Harrastustoiminnan aikana on tullut tehtyä useampia itsenäisiä pieniä julkaisuja, joita muut kanssaharrastelijat ovat kommentoineet. Lisäksi koulu on tarjonnut pelikehitys- ja suunnittelukursseja sekä mahdollisuuden opiskella lähes vuoden Oulu Game Labin parissa. Pelien analysointi niitä pelatessa vuosien aikana on myös edesauttanut tunnistamaan, mitä pelit tekevät oikein ja väärin sekä miksi. Kirjoittajan päätelmät eivät siis ole tuulesta temmattuja. Raportti sisältää myös ammattitermejä ja näitä on asiayhteydessä hiukan avattu.

2 JATKO-OSAN LÄHTÖKOHDAT

Tässä osassa käydään läpi, mitä otetaan huomioon, kun aletaan tehdä jatko-osaa videopelille. Päätöksellä tehdä jatko-osa on kuitenkin vaikutuksensa suunnitteluprosessiin. Samalla käydään läpi myös mahdollisia syitä jatko-osan teolle ja mitä hyötyjä tai haittoja päätöksestä voi seurata.

2.1 Miksi jatko-osa?

Videopelin ollessa menestys, tulee helposti ajatus jatko-osan kehittämisestä. Tämä on voinut olla oikeastaan jo suunnitelmissa ennen ensimmäisen pelin tekoa. Outsoftin blogitekstin mukaan (2015, viitattu 12.6.2016) pelin työstäminen jatko-osissa voi olla myös tuotannollisesti hyvin järkevää, vaikkakin potentiaalisesti tylsää ja yleisön vihaamaa puuhaa. Vihalla viitataan nykyiseen trendiin, jossa uusia sarjoja tehdään riskien takia vain harvoin ja iso osa peleistä on yleensä jatko-osa aiemmin menestyneelle sarjalle.

Kaiken taustalla on yksi selkeä motivaatio: raha. Ensimmäinen peli on jo tuonut rahaa, joten jatko-osa tulisi myymään jopa kahta peliä (Leonov 2014, viitattu 12.6.2016). Tällä tarkoitetaan siis sitä, että jatko-osa myydään ensimmäisen pelin maineella. Sarjaa tuntemattomat pelaajat todennäköisemmin päätyvät ostamaan kummatkin osat, jolloin myyntituloja voidaan saada kahdesta pelistä.

Kysynnällä on myös oma osansa jatko-osan teon motiiveihin ja se kulkeekin käsi kädessä rahan kanssa. Se ei silti ole yhtä varma motivaatio kuin raha. Esimerkiksi kuuluisa Half-Life 3 olisi varmasti kehittäjilleen valtava rahasampo, mutta peli edelleenkin loistaa poissaolollaan. Kuten Patrick Danen tekstistä (2014, viitattu 12.6.2016) käy ilmi, jotkut kehittäjät mieluummin arvailevat pelaajien tulevia mieltymyksiä sen sijaan, että kuuntelevat mitä nämä haluavat. Syy miksi näin haluttaisi tehdä on taas täysin subjektiivista. On silti selvää, että mikäli jatko-osalle on kysyntää, sille löytyy ostajia ja näin ollen mahdollisuus tehdä rahaa. Kannattavuus on tietenkin asia erikseen.

Taiteellisella ilmaisulla on myös roolinsa, kun puhutaan motivaatiosta jatko-osan tekoon. Se on ensiksikin uniikki mahdollisuus korjata aiemman pelin viat (Leonov 2014, viitattu 12.6.2016). Toiseksi, kun peli on suunniteltu tehtävän jatko-osissa, ei synny niin helposti kehittäjien loppuunpalamista, ylitöitä ja heitteleviä aikatauluja, kun peliä ei ole yritetty pistää samaan pakettiin (Outsoft

2015, viitattu 12.6.2016). Hurjat ideat voi porrastaa lähtemällä ensimmäisessä osassa kevyemmistä ideoista, lisätä niitä jatko-osissa, jolloin viimeinen osa sisältäisi kaikki herkut. Tarinalliset pelit voi myös rytmittää paremmin, mikäli se on jaettu osiin. Aiemman pelin jättämät kysymykset voivat myös luoda miljöön ja näin ollen motiivin jatko-osalle.

Loppujen lopuksi motiivit videopelin jatko-osan teolle eivät eroa jatko-osien teosta muilla luovan teollisuuden aloilla. Raha, kysyntä ja taiteellinen ilmaisu ovat aktiivisia motiiveja myös esimerkiksi elokuvateollisuudessa. Ihanteellisimpana motiivina voisi pitää yhdistelmää, jossa on vähän kaikkea.

2.2 Ero uuden pelin ja jatko-osan suunnittelun välillä

Ero uuden pelin suunnittelun ja jatko-osan välillä on pääasiassa siinä, miten lähdetään liikkeelle. Uuden pelin suhteen lähdetään aika tyhjästä: kohderyhmää ei tunne kunnolla (jos ollenkaan) ja mekaniikkojen toimivuudesta ja hauskuudesta ei voi mennä takuuseen. Tästä syntyvän riskin vuoksi onkin tärkeää uutta peliä tehdessä, että mekaniikkoja päästään testaamaan varhain. Mitä aiemmin testaa, sitä aiemmin löytää vikoja ja sitä varhaisemmin ne viat voi korjata ja tätä kautta voi voittaa pelaajien luottamuksen (Helppi 2013, viitattu 14.6.2016). Suunnittelun suhteen tämä voi taas tarkoittaa sitä, että suunnitelmia muutetaan hyvin usein. Jos pelin lajityyppi on myös uusi tuttavuus, syntyy tästäkin työtä: on ensin tunnettava mitkä ovat lajityypin nykyiset vaatimukset, eli mitä peliltä vähintään odotetaan mekaniikoiltaan. Voisi siis väittää, että uuden pelin suunnittelu on isompi ja vaikeampi työ. Tämä johtuu juuri siitä, että ei ole pohjalla mitään mitä käyttää.

Kun taas lähdetään suunnittelemaan jatko-osaa, on paljonkin mitä hyödyntää ensimmäisestä pelistä: pelin idea, dokumentaatio, pelin saama palaute, reaktiot, syntynyt maine sekä ilmenneet ongelmat tuotannossa sekä suunnittelussa. Mikäli jatko-osasta ei päättä tehdä aivan erilaista peliä, jatko-osan suunnittelu on siis teoriassa vähän helpompi työ. On myös epätodennäköistä, että suunnitelmia joudutaan muuttamaan yhtä usein kuin uutta peliä tehdessä, sillä testaamisella tässä vaiheessa on harvemmin suunnitelmia muuttavaa vaikutusta. Ideahan on jo todettu aiemman pelin myötä toimivaksi. Voi silti olla, että valmiiden materiaalien, kilpailun ja palautteen analysointi on hyvin työlästä, mutta tämäkin on tapauskohtaista. Mikäli pohjatyötä on silti tehty, se on merkittävä osa urakasta pois.

Tiivistetysti uusi peli suunnitellaan tyhjästä, mikä vaatii enemmän työtä. Jatko-osan suunnittelu ei välttämättä ole yhtään pienempi työ, mutta sen suunnittelua varten löytyy paljon pohjamateriaalia, jota käyttää hyväksi ja jolla helpottaa työtä.

2.3 Suunnittelun lähtökohdat

Seuraavaksi käydään tarkemmin läpi, mitä kannattaa ottaa huomioon jatko-osaa suunnitellessa. Osion selventämiseksi se on jaettu kolmeen osaan, joista ensimmäisessä käydään palautteen, kohderyhmän ja kilpailun merkitystä läpi yhdessä. Tämän jälkeen tulee osio jatko-osan samankaltaisuudesta ja ideoiden kierrätyksestä. Lopuksi on yhteenveto, jossa tiivistetään aiempien osien asioita.

2.3.1 Palaute, kohderyhmä ja kilpailu

Kun lähtee suunnittelemaan jatko-osaa, on tärkeää tutkia ensimmäisen pelin saamaa palautetta. Tämä siksi, koska palautteen kautta voi ymmärtää, mitä peli teki oikein ja mitä se teki väärin. Peliä pelanneet tuntemattomat ihmiset ovat parhain lähde kritiikille, koska he ovat ankarampia ja antavat rehellisempää palautetta. Jos saa palautetta henkilöltä jonka tuntee, tämä voi antaa anteeksi joitain asioita suhteenne takia. Sympaattisuus on tietenkin palautteessa mukavaa, mutta toisaalta rehellisestä palautteesta hyötyy paljon enemmän.

Tärkeintä on silti muistaa suhtautua hieman kriittisesti palautteeseen ja tasapainotella positiivisten ja negatiivisten kritiikkien välillä, nimittäin peliä pelanneilla on erilaisia mielipiteitä vähintään yhtä paljon kuin itse pelaajia. Mikäli yrittää miellyttää kaikkia palautteen antajia, lopulta ei todennäköisesti miellytä ketään. On tärkeää hyväksyä, että kaikkia ei voi eikä tule miellyttää, tunnistaa mitä pelillä halutaan tehdä ja mistä varsinainen kohderyhmä pitää/piti.

Hyvä esimerkki palautteen kanssa tasapainoilun suhteen on Dark Souls -pelisarja, joka tunnetaan hyvin siitä, että sarjan pelit ovat haastavia. Sarjaa rakastavat myös pitävät sarjasta juuri tästä syystä. Tämä on silti synnyttänyt ajatuksia siitä, että miksei pelissä ole helpompaa vaikeustasoa muita pelaajia varten. Esimerkiksi Oliver Cragg kertoo kirjoituksessaan (2016, viitattu 13.6.2016), että sarjasta voisi tulla ajattomampi sekä paljon tulkinnanvaraisempi, mikäli kokemus ei koostuisi

puhtaasta vaikeustasosta ja useampi pystyisi peliä pelaamaan. Mitä ikinä tästä onkaan mieltä, sarjan kehittäjät eivät koskaan tehneet helpompaa vaikeustasoa ja vaikkei tälle suoraa syytä ole koskaan kerrottu, tämän voidaan päätellä olevan taiteellinen päätös sekä halu miellyttää ydinkohde-ryhmää. Sarjan ohjaaja Hidetaka Miyazakin haastattelussa (Jenkins 2012, viitattu 14.6.2016) tämä kertoo haluavansa tehdä pelistä tyydyttävän kokemuksen pelaajalle ja vaikeustasolla on tarkoitus saada tämä aikaan.

Leonovin tekstin mukaan (2014, viitattu 12.6.2016) on hyvin tärkeää ensisijaisesti tuntea kohde-ryhmänsä. Tästä onkin suuri apu palautteen läpikäynnissä. Kun tietää kohderyhmän tottumukset ja mieltymykset, pystyy korottamaan tiettyjen palautteiden merkitystä sekä huomaamaan mitkä ovat ne oleelliset mekaniikat, jotka kaipaavat hiomista. Luonnollisesti kohderyhmä kannattaa tuntea myös ensimmäistä peliä tehdessä. Oman kohderyhmänsä pääsee tuntemaan vain puhtaalla markkinatutkimuksella, mutta se on aiheena sen verran laaja, ettei sitä tässä opinnäytteessä tulla käymään läpi.

Kohderyhmän lisäksi on hyvä tuntea myös kilpailijat markkinoilla. Kilpailijalla tarkoitetaan siis toista tahoa, joka tarjoaa vastaavaa tuotetta/palvelua. Tässä tapauksessa kilpailuksi luokituu ne tahot, jotka tarjoavat tai ovat tarjoamassa samankaltaista peliä. Kilpailijoiden ja heidän tarjontansa tunteminen voi auttaa erottamaan oman tuotteen ja palvelut muista (Info Entrepreneurs 2009, viitattu 14.6.2016). Jatko-osan suunnittelun kannalta onkin hyvin tärkeää tietää mitä pelejä markkinoilla on, sillä kilpailijat ovat voineet keksiä jotain, mistä syntyy uusi vähimmäisvaatimus pelin lajityypissä. Tällä siis vältetään se, että peli olisi epämuodikas jo ennen julkaisua. Kilpailulla on myös luontainen positiivinen vaikutus suunnitteluun, sillä markkinoilla kaikki tarkkailevat toisiaan ja yrittävät olla parempia kuin kilpailijansa. Toisin sanoen ei tyydytä siihen mitä on, vaan yritetään mennä vielä pidemmälle. Tämä yleensä johtaa entistä parempaan peliin. Mainittakoon myös, että vaikka puhutaan kilpailusta, nykyinen peliala on niin pieni, että kilpailijat todennäköisesti tapailevat toisiaan pelitapahtumissa ja antavat toisilleen tukea ja kommentteja. Kilpailijoiltaan voi siis saada myös asiantuntevaa palautetta.

Sanallisen palautteen lisäksi on nykyään mahdollista myös kerätä palautetta peliin kytkettyjen analyysijärjestelmien kautta. Kehittäjät voivat päättää näiden avulla missä kohtaa peliä ja minkälaista dataa peli lähettää. Jos järjestelmää hyödynnetään oikein, lopulliset analyysitulokset kertovat hyvin

paljon siitä, miten peliä pelataan. Tulokset eivät toki kerro kaikkea, mutta auttavat suuresti selvittämään pelaajien ajankäyttöä pelin parissa. Tämän pohjalta voidaan lähteä selvittämään syitä esimerkiksi sille, miksi pelaajat jäävät jumiin tiettyyn kohtaan pelissä.

2.3.2 Samankaltaisuus ja ideoiden kierrätys

On tärkeää, että jatko-osa säilyttää tarpeeksi samanlaisia elementtejä edeltäjästään, jotta se pystyy säilyttämään alkuperäisen kohderyhmänsä (Hartup 2014, viitattu 13.6.2016). Uusien ja vanhojen ideoiden välille on löydettävä tasapaino. Yleensä mekaniikat, säännöt ja järjestelmät peleissä ovat uniikkeja ja jos niitä lähtee radikaalisti muuttamaan tai jopa poistamaan jatko-osassa, aiemman pelin fanit eivät tunne enää vetoa peliin (2014, viitattu 13.6.2016). Tärkeää on silti tietää, mikäli asiat olivat pelissä oikeasti hyviä tai hauskoja ja miten ne vaikuttavat kokonaisuuteen. Sen perusteella on sitten päätettävä mitä ottaa pois, mitä korjata ja mitä ehkä tehdä lisää. Tämän voi selvittää aiemmin mainitun palautteen kautta tai jopa itse, jos uskoo kykenevänsä antamaan järkevää kritiikkiä omalle tuotokselleen.

Hartupin mukaan (2014, viitattu 13.6.2016) sarja ei saisi myöskään tarjota vain samaa vanhaa versiota pelistä. Erityisesti yksinpeleillä on suuri riski, että kauhu muuttuu tuttavaksi, jännitys arjeksi ja ilmiömäinen tavanomaiseksi. Vasta-argumenttina voisi toki mainita Call of Duty, NHL ja Assassin's Creed sarjojen menestykset, joissa ydinpeliä muutetaan vuosittain vähän ja silti ylläpidetään suuria tuloja. Metro Newsin lukijan Frankyn mukaan (2015, viitattu 12.3.2016) tämä voisi johtua siitä, että parantelu tavoitteena useimmin tuottaa paremman pelin kuin puhdas innovaatio. Toisin sanoen kyseessä on siis aiemmin mainittua kohderyhmän tuntemista ja lienee väärin myös vähätellä kyseisten sarjojen markkinointibudjettia, jolla uusia pelaajia on saatu mukaan. Aivan kuka tahansa ei pysty samaan.

Uuden ja vanhan tarjoaminen on silti myös tasapainotettava. Hyvässä pelissä yleensä viehättää jokin tietty asia, joten aivan heti ei ole mitään syytä muuttaa kaikkea kierrätyksen pelossa. Vaan sekään ei tarkoita, että kaikkea tulisi kierrättää. Jos pelaajat tykkäsivät esimerkiksi pelin kartasta, ei karttaa kannata laittaa sellaisenaan jatko-osaan. Tämä luo vaikutelman siitä, että suunnittelussa on oltu laiskoja. Tämä ei myöskään kannusta pelaamaan, sillä tämä kartta on jo tutkittu, kartassa ei ole uutuudenviehätystä. Myös pelissä kohdatut vastukset ja pomotaistelut vaikuttavat heti tylsiltä, jos nämä muistuttavat selkeästi aiemman pelin vastikkeita.

Selvää kierrätystä voi toki käyttää eduksi ja joissain tilanteissa se on oikeutettua. Uusi kartta voi esimerkiksi sisältää pienen muunnellun palasen aiemmasta kartasta. Pelaaja muodostaa täten yhteyden alueeseen nopeasti aiemman pelin kautta. Tämän mielenkiinto voi nousta entisestään, sillä tämä todennäköisesti haluaa etsiä eroja ja mahdollisia salaisuuksia edukseen, joita aiemmassa pelissä oli. Alueeseen voi myös liittyä nostalgiaa. Tätä ei voi silti tehdä sellaisenaan, vaan tätä ennen pelin on vakuutettava pelaaja siitä, että alueisiin on nähty vaivaa, muuten tämä voi tuntua pelaajasta laiskuudelta.

Hyvä esimerkki pelistä, joka tietoisesti kierrättää kaiken ja tekee sen vielä fiksusti, on Metal Gear Solid 2. Kyseessä on yhdistelmä pelimekaanista uusioversiota sekä kerrontaa tarinassa: alue ja asiat joita teet ovat erilaisia, mutta silti erittäin samankaltaisia ensimmäiseen peliin verrattuna. Tekniset erot ovat kuitenkin merkittäviä ja pelaaminen tuntuu tästä syystä aivan erilaiselta. Lisäksi kierrätykselle löytyy myös selityksensä tarinan kautta. Pelaaja ei edes välttämättä tajua kierrätystä aiemmin mainittujen teknisten erojen takia, ennen pelin loppupuolella tapahtuvaa juonenkäännettä.

Asioiden kierrätys on siis hankala asia pelisuunnittelussa, sillä on helppo saada jälki vaikuttamaan laiskalta. Käytännössä jos katsoo mitä tahansa peliä korkeammalla tasolla, niin perusideoita kierrätetään jatkuvasti. Hyvänä nyrkkisääntönä voisi siis olla, että näkee vaivaa ja näyttää sen, oli idea kierrätetty tai ei. Pelin sisältämät vastustajat voivat esimerkiksi käyttäytyä samalla tavalla, mutta mikäli näillä on uusi ulkoasu, uusi animaatio ja uusi tarina takanaan, tähän on nähty vaivaa ja sen saa tuntumaan uudelta. Mikäli kierrätyksestä muodostuu tunnusomainen piirre sarjan saamassa palautteessa, on syytä pohtia uuden pelin tekoa jatko-osan sijaan. Jos yleisö väsyi, tämä ei todennäköisesti osta seuraavaa osaa.

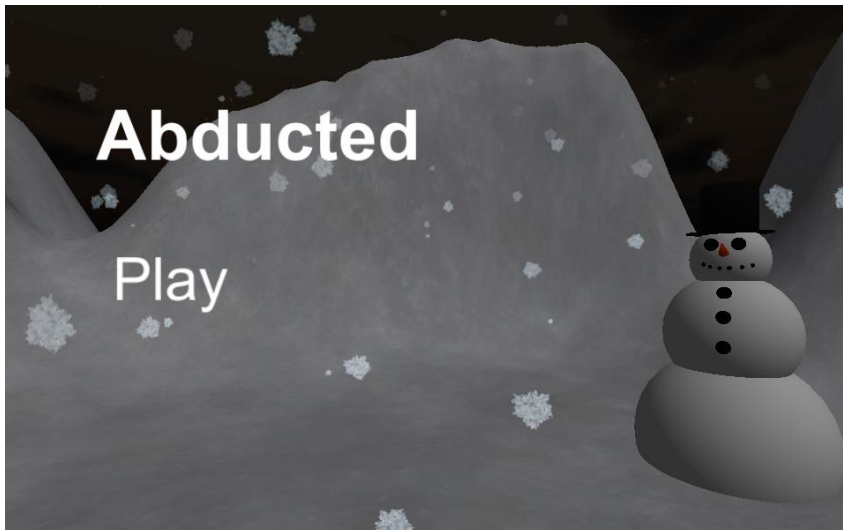
2.3.3 Yhteenveto

Loppupäätelmänä voidaan todeta, että jatko-osaa alettaessa suunnitella on hyvä huomioida, ensimmäisen pelin saama palaute, pelin kohderyhmä ja pelin kilpailijat. Suunnittelussa on taas hyvä huomioida uusien ja vanhojen ideoiden tasapainottaminen niin, että pelin kohderyhmää ei karkota eikä pelissä ala maistua kierrätys. Jatko-osan suunnittelu on siis hyvin paljon ensimmäisen pelin

vahvuuksien ja heikkouksien analysointia sekä tasapainoilua ideoiden ja ajatusten kanssa. Ensimmäinen peli jättää jälkeensä kuitenkin paljon materiaalia hyödynnettäväksi, mutta näiden tulkitseminen voi viedä oman aikansa.

3 ENSIMMÄINEN PELI

Tässä osiossa käydään läpi peliä, jolle jatko-osa tehdään. Aluksi on tarkoitus selvittää mistä pelissä on kyse ja siirtyä sitten määrittelemään, miten peliä lähdetäisiin jatkokehittämään.

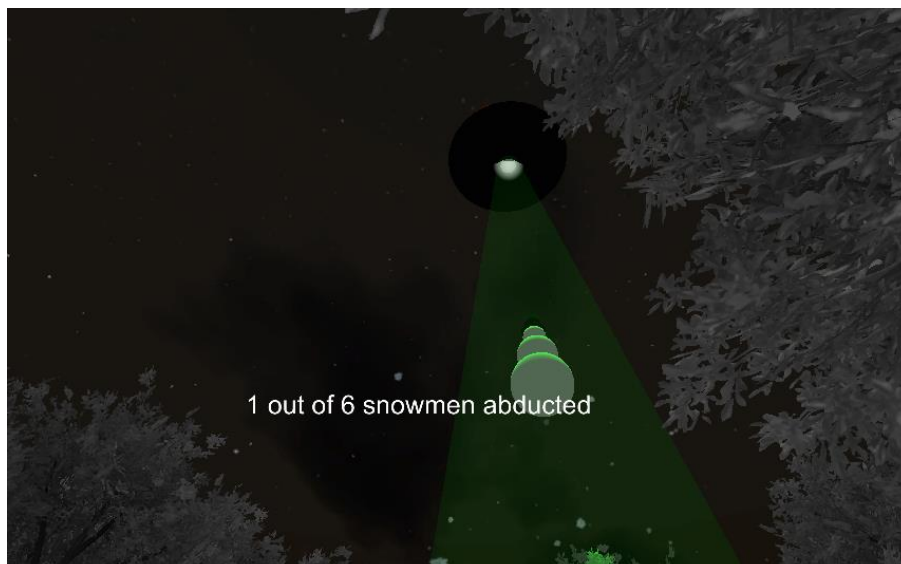


Kuvio 1 Pelin Abducted alkuvalikko

Peli on nimeltään Abducted. Kyseessä on 2014 syksyllä tehty toimintapeli. Peli tehtiin viikossa Unity 3D -ohjelmistolla, joka on ilmainen pelikehitystyökalu. Peliä pystyy kirjoitushetkellä pelaamaan ilmaiseksi Game Jolt -sivustolla suoraan selaimesta, mutta tämä vaatii Unity Web Player -selainliitännäisen asentamista. Tämä kyseinen liitännäinen mahdollistaa sen, että Unity 3D -ohjelmistolla tehdyt Web Player -koontiversiot toimivat suoraan esimerkiksi Mozilla Firefoxissa. Tämä on nykyään vanhentunutta tekniikkaa ja esimerkiksi Google Chrome ei tue tätä liitännäistä ollenkaan. Peli ei ole siis pelattavissa kaikilla selaimilla.

3.1 Pelin yleiskuvaus

Kuvaus pohjautuu pelistä annettuihin tietoihin pelin sivulla ja kokemuksiin parin pelikerran jälkeen. Abducted on kolmiulotteista grafiikkaa sisältävä peli ja se on kuvattu ensimmäisestä persoonasta. Peliä pelataan liikkumalla ja katselemalla ympäriinsä. Katseluun ja liikkumiseen käytetään pelin sivun mukaan WASD-näppäimiä ja hiirtä (Game Jolt 2014, viitattu 10.6.2016).



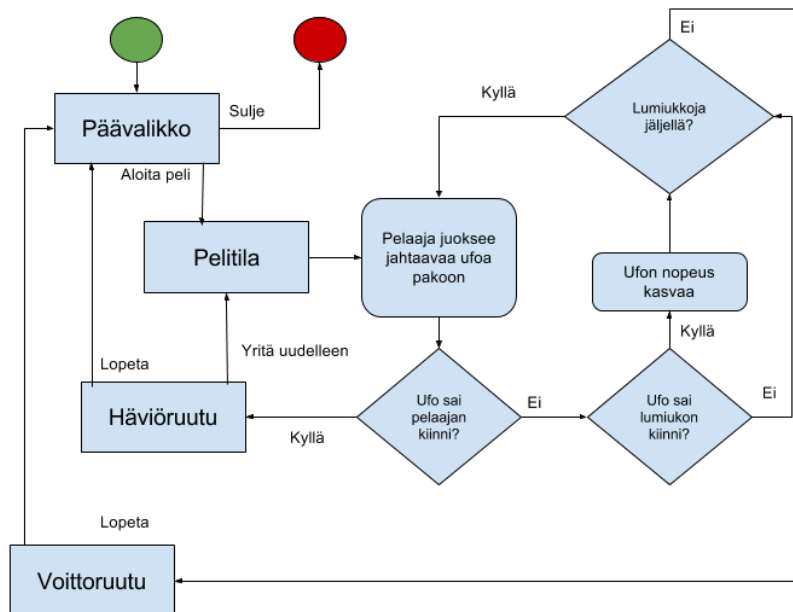
Kuvio 2 Tilannekuva itse pelistä

Pelin alkaessa pelaaja heitetään talviseen ympäristöön, jossa pelaaja näkee vain puita, lunta ja liikkuvia lumiukkoja. Pelaajalle ei esitetä pelissä minkäänlaista selkeää tavoitetta, mutta pelaajan on tarkoitus juosta pakoon tätä jahtaavaa ufoa ja johdattaa tämä nappaamaan lumiukot itsensä sijaan. Ufon nopeus kasvaa jokaisen napatun lumiukon myötä ja mikäli tämä saa pelaajan kiinni, tämän pitää aloittaa alusta. Lumiukkojen loppuessa peli onnittelee pelaajaa petturuudesta, vihjaten pelaajan itsensä olevan myös lumiukko.

3.2 Takaisinmallinnus

Koska pelistä ei ole olemassa minkäänlaista dokumentaatiota, sen toiminnallisuutta voidaan selvittää takaisinmallintamalla (reverse engineering). Tällä tarkoitetaan laitteen tai järjestelmän toiminnallisuuden selvittämistä, eli mitä se tekee ja miten. Tätä voi lähestyä usealla eri tavalla, kuten analyysillä silmä- ja kokemusmääräisesti, tai purkamalla ohjelman osiin niin, että voit lukea esimerkiksi ohjelman alkuperäistä koodia.

Pelin rakennetta voisi kuvata seuraavalla kaaviolla kuviossa 3. Peli käynnistyy vihreässä pallossa ja sulkeutuu punaisessa.



Kuvio 3 Kaavio pelin toiminnallisuudesta

Kuten kuvio 3 käy ilmi, peli koostuu neljästä tilasta/ruudusta (kuvattu neliöinä) ja pelin toiminnallisuus vastaa aiemmin tehtyä pelin yleiskuvausta. Huomattavaa myös on, että kyseessä ei ole millään muotoa karkea kuvio, peli nimittäin on todellakin rakenteeltaan näin yksinkertainen.

Seuraavaksi yritetään saada peliä esittävä ohjelma auki ymmärrettävään muotoon ja tutkia sen rakenteita takaisinmallintamalla. Tällä tavalla voidaan huomata teknisiä kehityskohteita sekä ymmärtää paremmin pelin toiminnallisuutta. Abductedin suhteen tämä on oikeastaan turhaa, sillä alkuperäinen koodi, äänet ja grafiikka ovat tallessa. Tyypillisesti takaisinmallinnusta hyödynnetään, kun projektin tiedostoihin ei pääse käsiksi ja tästä ei tässä tapauksessa ole kyse. Tämä on silti mielenkiintoinen aihepiiri, jota käsitellään pintapuolisesti. Unity 3D -ohjelmalla tehdyille peleille tämä on myös helppoa tehdä.

Tavanomaisesti ohjelmistokehityksessä kirjoitettu koodi käännetään koontiversioksi, joka taas jaetaan asiakkaalle. Koontiversiolla tarkoitetaan tiettyä käännettyä versiota tietokoneohjelmasta, jota voi myös oikeastaan kutsua valmiiksi tietokoneohjelmaksi. Tämän sisältämät osat ovat yleensä pelkkää konekieltä tai salattua tietoa, joten edes kokenut ohjelmoija ei pysty tietokoneohjelman tarkkaa rakennetta tulkitsemaan ilman alkuperäistä koodia. Se ei silti ole mahdotonta, sillä jos tietää millä tietokoneohjelma on tehty, kokeneempi ohjelmoija pystyy päättämään ohjelman rakenteita ja näin ollen kirjoittamaan oman ohjelman, jolla ohjelman voi purkaa.

Menetelmä voi kuulostaa laittomalta, mutta mikäli prosessi jää vain ohjelman tutkimiseen tässä ei ole mitään väärää. Olli Sulopuiston artikkelin mukaan (2012, viitattu 10.6.2016) EU-tuomioistuin teki erään takaisinmallinnukseen liittyvän keissin suhteen päätöksen, että takaisinmallinnus itsessään ei riko tekijänoikeuslakeja. Sen sijaan, jos takaisinmallinnuksesta saatuja osia varastaa omaan käyttöön ja tekee tällä rahaa, tämä voidaan luokitella rikolliseksi toiminnaksi. Tapauskohtaisesti voi ilmetä poikkeuksia.

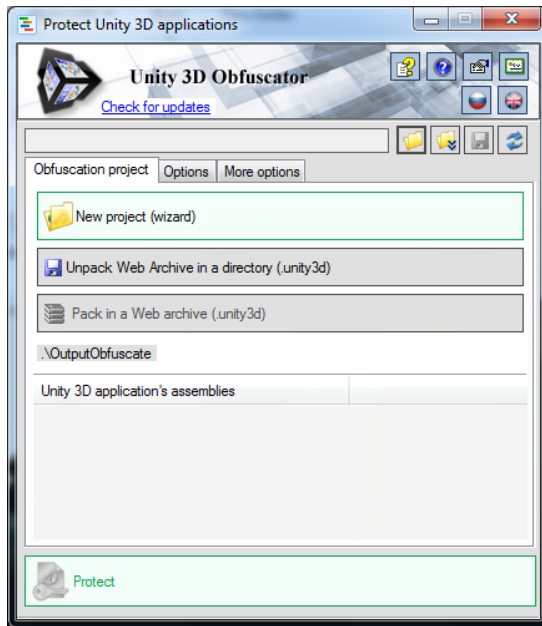
Alan Zucconin kirjoittamassa artikkelissa (2015, viitattu 10.6.2016) kerrotaan aika tyhjentävästi, miten Unity 3D -ohjelmistolla tehdyn koontiversion saa purettua ja miten purkamista voi mahdollisesti estää. Tässä opinnäytteessä purkamisen estoja ei käydä mitenkään läpi, mutta aiemmin mainitun artikkelin pohjalta on otettu oppia seuraavissa teksteissä esiintyviin purkuyrityksiin.

Abductedista on olemassa alkuperäisen Unity 3D -projektin lisäksi Unity Web Player -koontiversio. Tämä poikkeaa perinteisestä ohjelmasta siinä, että tätä koontiversiota ei voi käyttää sellaisenaan, toisin kuin esimerkiksi PC:lle tehtyä koontiversiota. Se on ensin upotettava nettisivulle, jolla se taas toimii, mikäli omaan selaimeen on asennettu Unity Web Player -liitännäinen. Tällaisen koontiversion purkamiseen voidaan käyttää kolmea ohjelmaa: Unity 3D Obfuscator, ILSpy sekä Unity Assets Explorer. Seuraavaksi käydään näitä ohjelmia lyhyesti läpi Abductedin purkamisen merkeissä, minkä jälkeen tiivistetään ajatuksia tämän takaisinmallinnuksen pohjalta.

3.2.1 Unity 3D Obfuscator

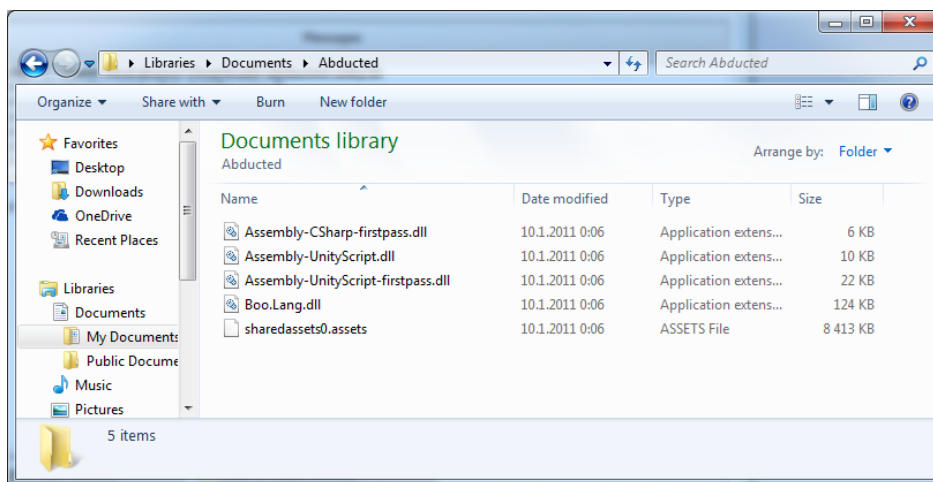
Unity 3D Obfuscator on ainoa tuote, joka tarjoaa ammattimaiset keinot suojata Unity 3D:llä tuotetut ohjelmistot väärinkäyttöä ja muuntelua vastaan (Tech-Next 2010, viitattu 10.6.2016). Ironista kyllä, ohjelma antaa myös mahdollisuuden purkaa Web Player -koontiversioita osiin, joista pääsee käsiksi pelin koodiin, ääniin ja grafiikkaan.

Tätä ohjelmaa ei tarvita, mikäli haluaa purkaa esimerkiksi PC:lle tehdyn koontiversion. Ohjelman suojaustoimintoja ei tässä käydä läpi ja purkutoiminta käydään läpi pinnallisesti.



Kuvio 4 Unity 3D Obfuscator -ohjelma

Kuviosta 4 näkee miltä Unity 3D Obfuscator näyttää. Käyttö on myös hyvin yksinkertaista: painamalla "Unpack Web Archive in a directory (.unity3d)" voit etsiä halutun Web Player -koontiversion, valita kansion mihin haluat tämän purettavan ja hetken odottelun jälkeen purku on tehty. Abducte-din kohdalla tuli vastaan virheviesti, mutta tämä ei näyttänyt vaikuttavan mihinkään, kuten kuviosta 5 voi huomata. Voi kyllä olla, että myöhemmin ilmenevällä virheellä on tämän kanssa jotain tekemistä.

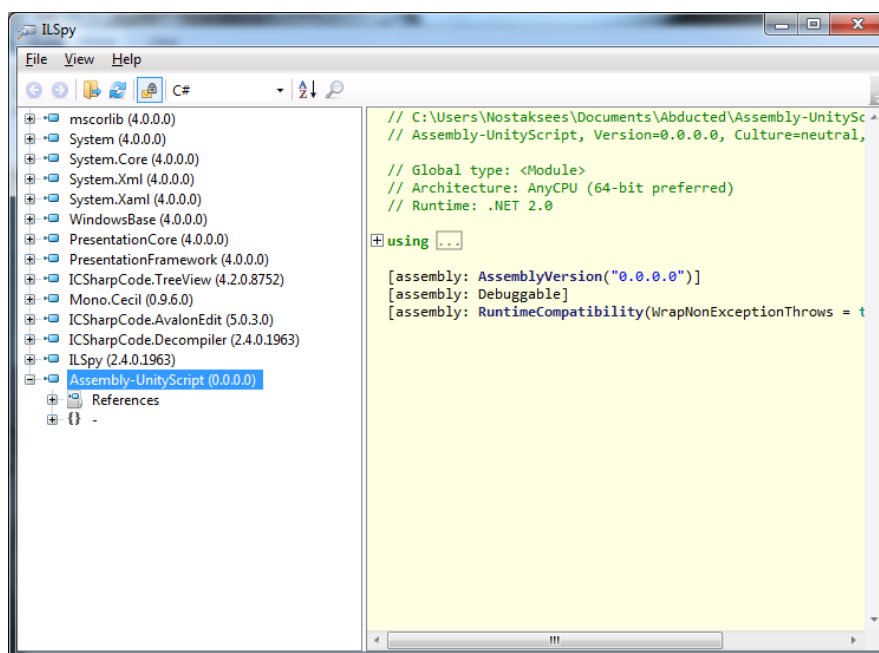


Kuvio 5 Purettu peli ohjelman käytön jälkeen

Ohjelma purki Unity Web Player -koontiversion siis onnistuneesti. DLL-tiedostot sisältävät pelin koodin ja assets-tiedostot taas pelin assetit. Assetilla tarkoitetaan ääntä, grafiikkaa ja fontteja; periaatteessa siis niitä palasia joilla peli rakennetaan. Näitä ei voi silti tällaisinaan vielä tulkita. Mainittakoon, että nämä tiedostot löytyvät esimerkiksi PC:lle tehdyssä koontiversiossa pelin Data-kansioista.

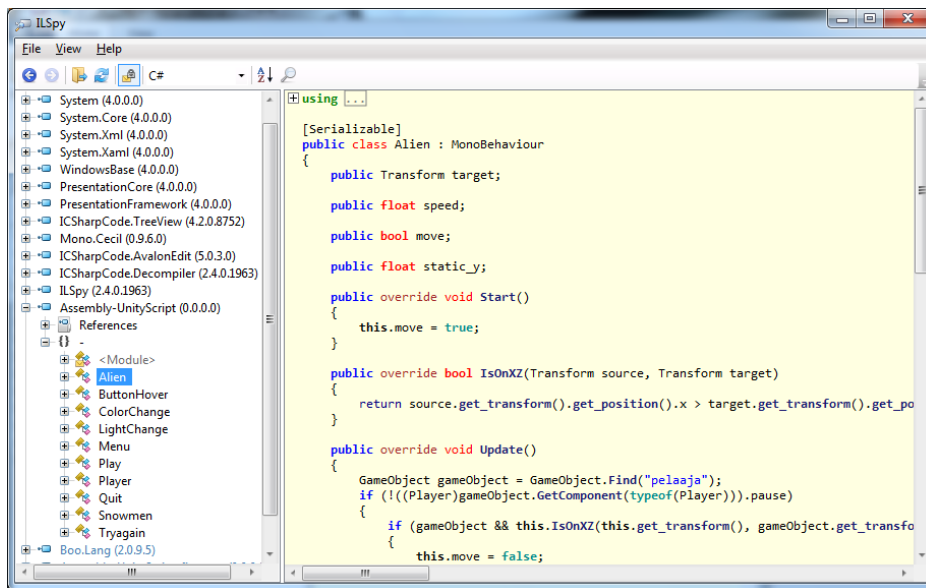
3.2.2 ILSpy

Tämä avoimen lähdekoodin ohjelma mahdollistaa .NET kokoonpanojen koodien selaamisen sekä hajottamisen osiin (Icsharpcode 2016, viitattu 10.6.2016). Koska kyseessä on siis .NET-pohjainen ohjelma, ei sitä voida käyttää kuin Windows-käyttöjärjestelmällä ja vain sille tehtyihin ohjelmiin. Tässä tapauksessa vaatimukset täytetään.



Kuvio 6 ILSpy-ohjelma

Oletuksena ohjelma vaikuttaisi jo löytävän automaattisesti paljon selattavia osia, mutta tällä hetkellä oleellista on vain Abductedin koodi. Painamalla keltaista kansio kuvaketta vasemmasta yläkulmasta saa etsittyä aiemmin tuotetut dll-tiedostot ja ohjelma lisää näiden sisällöt samaan listaan minkä oletuksena näkee. Tämän jälkeen näiden sisältämää koodia voi tutkia.



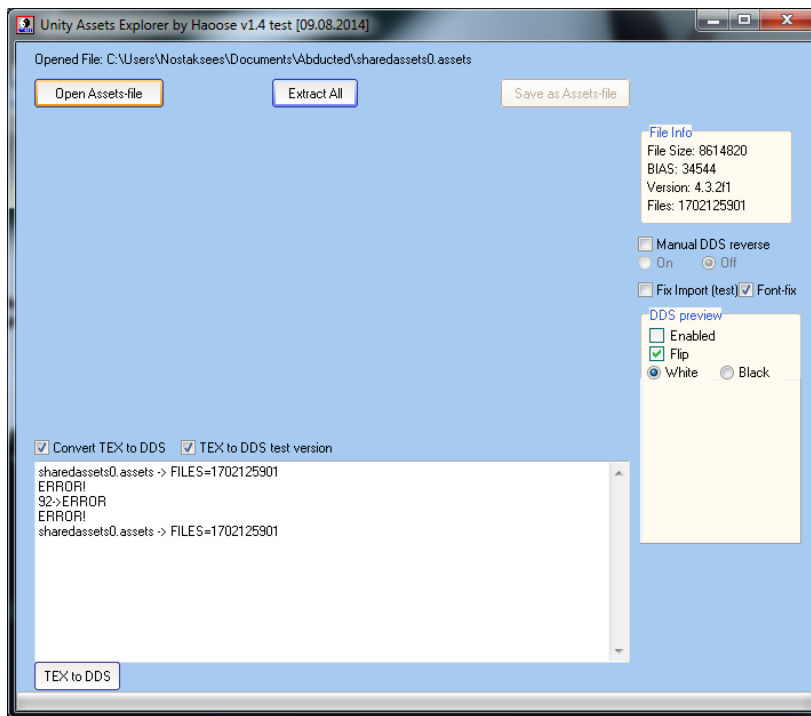
Kuvio 7 Koodin tarkastelua

Mielenkiintoista on huomata tässä välissä, että vaikka alkuperäinen koodi kirjoitettiin UnityScript-ohjelmointikielellä, koontiversion tiedostoissa tämä on muunnettu silti C#-ohjelmointikielelle. Näillä kahdella ohjelmointikielellä ei ole kuitenkaan muuta oleellista eroa kuin syntaksi, eli miten koodia kirjoitetaan. Koodista myös huomaa selkeästi, että koontiversion koodissa Unity 3D -ohjelmiston omaa syntaksia on paljon vähemmän.

Tässä vaiheessa kaikkea pelin koodia pystyy selaamaan ja kokenut ohjelmoija pystyy huomaamaan lukuisia asioita, mitä voisi parantaa jatkossa. Esimerkiksi: koodia on vähän ja se on kohtalaisen yksinkertaista, mutta suorituskyvyn kannalta se käyttää silti raskaita menetelmiä. Esimerkiksi GetComponent-funktiota on raskasta kutsua sen toiminnallisuuden vuoksi Unity 3D -ohjelmiston Update-funktiossa, koska tämä funktio käydään ohjelmaa ajaessa sen jokaisessa kehyksessä (Unity Technologies 2016, viitattu 10.6.2016). Sen palauttama arvo on siksi suositeltavampaa asettaa omaan muuttujaan esimerkiksi Start-funktiossa, jolloin funktio ajetaan vain yhden kerran ja näin ollen keventää ohjelman tekemää laskentaa. Kyseenalaistettavissa on myös menetelmä koodin kirjoittamisessa, jossa täsmälleen samalle asialle on tehty useampi eri koodinsa, vaikka tässä olisi voinut hyödyntää periittämistä taikka vain fiksumpaa ohjelmointia. Periittämisellä tarkoitetaan menetelmää ohjelmoidessa, jossa isä-lapsi rakenteella voidaan kirjoittaa samankaltaista koodia ilman, että samaa koodia tarvitsee kirjoittaa useampaan otteeseen.

3.2.3 Unity Asset Extractor

Tällä ohjelmalla pystyy purkamaan Unity 3D -ohjelmistolla tuotettujen koontiversioiden assets-tiedostoja, jotka sisältävät kaikki pelin assetit. Riippuen pelin koosta, näitä voi olla useampikin kuin yksi. Abductedin tapauksessa näitä on vain yksi.



Kuvio 8 Unity Asset Extractor -ohjelma

Ohjelman käyttö on yksinkertaista: painamalla “Open Assets-file” -nappia vasemmassa yläkulmassa, voit etsiä haluamasi assets-tiedoston, jonka jälkeen ohjelma miettii hetken ja antaa listan pelin sisältämistä aseteista. “Extract all” -painikkeella nämä assetit voi purkaa haluamaansa paikkaan. Abductedin tapauksessa tulokset jäivät hyvin laihoiksi: tuntemattomasta syystä ohjelma mietti ja jäättyi hyvin pitkiä aikoja eikä lopulta halunnut tuottaa minkäänlaista listaa pelin aseteista. Ohjelman joutui myös sammuttamaan tehtävienhallinnasta. Ohjelmasta jää hiukan vaikutelma keskeneräisyydestä ja puutteellisuudesta, mutta voi olla, että itse assets-tiedostossa oli jotain poikkeavaa. Aiemmin esiintynyt virhe Unity 3D Obfuscatorin kanssa saattaa liittyä tähän, mutta ongelmien epäselvyyden vuoksi tätä ei yritetty selvittää.

Näin ollen on, tämän takaisinmallinnuksen näkökulmasta, mahdotonta muodostaa tarkkaa kuvaa Abductedin sisältämistä aseteista. Joskin kun itse peliä pelaa, näiden vähäisyyden puolesta ei ole

vaikeaa tehdä karkeaa listaa. Peli käyttää Unity 3D -ohjelmiston tarjoamaa oletusfonttia Arial. Pelistä löytyy ääni kävelylle, ufolle ja tämän säteelle. Pelin 3D-mallit koostuvat lumiukosta, Unity 3D -ohjelmistosta valmiiksi löytyvästä puusta (johon vain väri on muutettu) sekä ufosta.

3.2.4 Päätelmät

Pelkällä pelaamisella saatiin selville paljon pelin toiminnallisuudesta ja sen rakenteesta, sekä puutteista. Vaikka assetteihin ei päästy käsiksi, peliä pelatessa pystyy jo näkemään paljon laadullisia puutteita: harmaa ja pimeä tyyli on tyydyttävä, muttei täysin yhtenäinen, kuten kuviosta 9 voi huomata.



Kuvio 9 Tilannekuvaa pelistä

Ufo ja lumiukot ovat hyvin yksinkertaisia ulkoasultaan vaan sen sijaan puilla on paljon yksityiskohtia ja jopa fysiikkaa. Äänimaailma on myös hyvin köyhä ja koostuu vain parista ääniefektistä. Pelin koodista kävi myös ilmi se, miten peliä ei ole optimoitu juuri yhtään ja koodin on selkeästi kirjoittanut joku, joka ei Unity 3D -ohjelmistoa hirveästi tunne.

Isoimmat puutteet ilmenevät suunnittelussa ja perusominaisuuksien puutteessa: ensimmäisen persoonan tietokonepelissä annetaan aina mahdollisuus pelaajalle säätää pelin asetuksia tälle sopivaksi, hiiren liikkuminen pitäisi rajata itse peliruutuun ja pelin tavoite on hyvin siinä rajoilla, että onko se liian epäselvä. Kokonaisuutta voi myös pitää turhan yksinkertaisena, pelin nimittäin peittoaa nopeasti eikä varsinaisessa ideassa ole syvyyttä. Näillä tiedoilla on jo hyvä lähteä liikkeelle jatkosan tai ylipäättänsä paremman version suunnittelussa.

3.3 Palaute

Abducted sai julkaisun, joten sille on tullut myös palautetta. Julkaisu tapahtui Game Jolt -sivustolla, jolle kuka tahansa voi rekisteröityä ja ladata oman pelinsä muiden pelattavaksi/ladattavaksi ilman kustannuksia. Nyt sivustolla voi myös myydä pelejään. Yhteisö on iso ja kasvaa jatkuvasti, joten markkinointikonstit ovat suureksi eduksi, mikäli haluaa sivustolla pelilleen näkyvyyttä.

Abductedin markkinointisuunnitelma oli, ilmaisuudestakin johtuen, hyvin yksinkertainen: kerro pelistä Internet-sivustoilla. Tavoitteena ei ollut lähteä valloittamaan jokaista foorumia vaan lähinnä kertoa pelistä niillä foorumeilla, joilla on ollut jo vakituinen käyttäjä. Rekisteröinti pelkän mainostuksen vuoksi on tavanomaisesti hyvin paheksuttavaa, joten kyseiseltä menetelmältä ei voisi odottaa palautetta. Näihin foorumeihin/sivustoihin, joilla lopulta Abductedia tuli esiteltä, kuului Pelaa-jalehti, TIGSource, Playfire ja Game Maker Suomi. Jokainen sivusto liittyy tavalla tai toisella video-peleihin. Vaikkei mistään pysty liikenteen статистиikkaa näkemään, Game Jolt -sivun kommentteista voi päätellä, että suurin osa pelikerroista on tullut itse Game Joltista.

Pelin saama palaute oli yllättävän positiivista. Yhtäkään kirjoitettua haukkua tämä ei ole saanut, sikäli kun on tiedossa. Paljolti pelin ideaa ja sen potentiaalia jatkoon suhteen keuhuttiin, monien tehden jopa ehdotuksia jatkoa varten. Parannuksia koettiin tarvittavan grafiikassa, asetuksissa ja muutamien pelimekaanisten ehdotusten muodossa. Pääasiassa palautteesta kävi ilmi, että kunnon jatkokehitystä toivottaisiin. Tämän pohjalta idea tälle opinnäytteelle syntyikin.

3.4 Jatko-osan lähtökohdat

Aiemmassa isossa osiossa käytiin läpi mitä lähtökohtia tulee ottaa huomioon jatko-osaa tehdessä. Tässä osiossa on tarkoitus käydä samoja asioita läpi, mutta Abductedin näkökulmassa. Lähtöasetelma on, että Abductedista tehdään jatko-osa. Motiivina on taiteellinen ilmaisu sekä kysyntä, sillä kuten aiemmassa osiossa läpikäydystä palautteesta kävi ilmi, pelille toivottiin jatkokehitystä. Jatko-osalla ei ole toistaiseksi rahoitussuunnitelmaa eikä edes varmuutta kunnon jatkokehitykseen. Tämän opinnäytteen tavoitteena on silti tehdä sille hyvä pohja.

Tarkoituksena ei ole siis määritellä perusteellisesti jatko-osan olemusta. Tarkoituksena on määritellä lähtökohtia, joiden pohjalta varsinainen suunnittelu aloitetaan.

3.4.1 Palaute ja ehdotukset

Palautteesta kävi ilmi paitsi se, että pelille toivottiin jatkokehitystä, myös se, että pelin idea oli hauska. Tätä ei ole siis syytä muunnella jatko-osaa varten, jollei keksi vielä jotain millä ideaa itseään rikastaa. Heikkoutena pidettiin sen sijaan graafista tyyliä. Vaikka tarkoituksena on tehdä vain prototyyppi, pelin nykyinen epäyhtenäinen ulkoasu olisi hyvä yhtenäistää. Toinen heikkous oli asetusten puute, mikä on ehdottomasti syytä korjata jopa tässä opinnäytteessä tehtävään prototyyppiin.

Siinä olikin nykyisen pelin saama kritiikki, loput palautteesta koostuu ehdotuksista. Näihin lukeutuu työkalut, ufon työkalut, moninpeli ja vaihtelevat tavoitteet. Seuraavaksi käydään näitä läpi ja pohditaan ovatko nämä toteutettavissa taikka edes järkeviä.

Työkaluilla tarkoitetaan tässä yhteydessä pelin kartalta löydettäviä esineitä, joita poimimalla saa käyttöönsä uusia ominaisuuksia. Tällainen voisi olla esimerkiksi köysipyssy, jonka avulla voi liikkua nopeammin paikasta toiseen. Tai vaikka vale-lumiukko, jonka asettaminen kartalle vie ufon huomion hetkeksi. Tällaiset työkalut tuntuvat hyvin luontevalta pelin jatkokehitystä ajatellen, sillä nämä tukisivat alkuperäistä ideaa. Ufolta pakeneminen voisi olla hauskenpaa, jos pelaajalla on enemmän hallintaa tilanteesta. Tämä ominaisuus sisällytetään paitsi suunnitelmiin, myös toteutettavaan prototyyppiin.

Ufon työkaluilla tarkoitetaan joko tietoisesti tai satunnaisesti aktivoitavia kykyjä, jotka helpottavat ufon toimintaa. Mikäli pelissä olisi esimerkiksi piilopaikkoja, voisi ufolla olla lasersäde jolla tuhota piilopaikkoja. Ideana tämä tuntuu työkalujen lailla hyvin luontevalta jatkokehitystä ajatellen, mutta nämä kyvyt voivat olla verrattaessa sen verran vaikeampia tehdä, että toteutettavassa prototyyppissä niitä ei nähtäisi.

Moninpelissä tarkoitus olisi taas antaa pelaajien kilpailla samassa asetelmassa: kuka selviää viimeisenä? Mahdollisuutena voisi antaa myös vaihtoehdot tekoäly-ufon sekä jonkun hallitseman ufon välillä, puhumattakaan yksinkertaisista tekoäly-vastustajista. Työkalut voisivat tukea myös tätä pelimuotoa: osa työkaluista auttaa omaa pelaamista ja osa taas vaikeuttaa muiden pelaamista. On kyseenalaista pystyisikö moninpeliä tekemään toimivaksi edes prototyyppiin, mutta tämä voisi silti olla yksi tärkeimmistä ideoista jatko-osan kannalta, sillä kuten palautteesta kävi ilmi, tällainen kukulan kuningas tyylinen peli kaipaasi moninpeliä.

Vaihtelevilla tavoitteilla tarkoitetaan, että pelissä vaihtuisi tavoite joksikin muuksi siitä, että lumiukkoja syötetään ufolle. Moninpelin myötä on kenties turha miettiä tätä, ainakaan prototyypin kanalta, sillä pelkkään prototyyppiin muiden pelaajien eliminointi on jo hyvin paljon tavoitteeksi.

Tässä kohtaa alkaakin jo syntyä selkeä kuva siitä, mitä Abductedin jatko-osalta halutaan. Tavoitteena tulee siis olla tehdä moninpeli. Pelin idea tukee tätä hyvin paljon. Moninpeliä taas elävöittäisiin antamalla mahdollisuus vaikeuttaa ja parantaa omaa peliään työkaluilla, joita löytyisi ympäristöstä satunnaisesti. Näiden mekaniikkojen lisäksi pelin ulkoasu tulisi olemaan yhtenäisempi ja peli tarjoaisi myös mahdollisuudet muokata pelin asetuksia pelaajalle sopivaksi.

3.4.2 Kohderyhmä

Koska suunnitelmassa on tehdä moninpeli, kohderyhmä verrattuna ensimmäiseen peliin tulee olemaan aivan eri. Ensimmäisellä pelillä ei myöskään varsinaista kohderyhmää oltu määritelty. On siis vähän vaikeampi ottaa aiempaa kohderyhmää huomioon näin jatkoa suunniteltaessa. On kuitenkin yksi asia, mikä voisi huolettaa: pelin vaihtuminen yksinpelistä moninpeliin. Voisiko joku aiemmista pelaajista närkästyä muutokseen? Mahdollisuus on olemassa, mutta on todennäköisempää, että idean toimivuus moninpelinä koetaan arvokkaampana, kuin heikompaa yksinpeli-toteutusta. Ajatuksen voi kyseenalaistaa, mutta tässä voidaan puhua juuri siitä, että alkuperäinen kohderyhmä hylätään uuden tilalle taiteellisen ilmaisun nimissä.

Kuten mainittua, valitettavasti minkäänlaista dataa ei löydy, jolla määritellä tarkalleen Abductedin kohderyhmää. Ensimmäinen pelikin on tehty oikeastaan sokkona ilman mitään ajatusta siitä, kenelle peli tehdään.

Vaikka kyseessä on opinnäyte ja työ tehdään itselle, seuraa nyt muutamia päätelmiä siitä, millainen pelin kohderyhmä olisi:

- Peli tulee olemaan alustavasti pelattavissa selaimessa, joten kohderyhmään kuuluu ihmiset jotka tykkäävät selainpeleistä.
- Koska peli tulee olemaan pelattavissa selaimessa, pelin laitevaatimukset eivät tule olemaan kovia. Kohderyhmään kuuluu siis ihmiset, joilla on kohtalaisen nykyaikainen tietokone.

- Peli tulee olemaan saatavilla Game Jolt -sivustolla, joten kohderyhmään kuuluu sivuston käyttäjäkunta. Sivuston käyttäjäkunta jakautuu vielä pienempiin osiin, mutta keskitytään nimenomaan niihin jotka pääasiassa pelaavat pelejä sivustolla.
- Peli tulee sisältämään moninpelin, joten kohderyhmään kuuluu pelaajat, jotka pitävät moninpeleistä. Tämän lisäksi kohderyhmään kuuluu pelaajat, joilla on hyvä Internet-yhteys.
- Peli tulee painottumaan hauskanpitoon ja tulee olemaan visuaalisesti lapsekas, joten kohderyhmään kuuluu pelaajat, jotka hakevat peliltä huvitusta ja ovat avoimia vähemmän realistisemman grafiikan suhteen.
- Peli tulee olemaan ilmainen, joten kohderyhmään kuuluu ihmiset jotka todennäköisesti pelaavat ilmaisjakelupelejä.

Esimerkki kohderyhmän edustajasta on siis henkilö, joka pelaa selainpelejä keskinkertaisella tietokoneella, omistaa hyvän Internet-yhteyden ja etsii ilmaisia pelejä saadakseen huvitusta Game Joltin kaltaisilta sivustoilta. Tämän profiilin pohjalta ei ainakaan heti ilmene asioita, joita tulisi ottaa huomioon suunnittelussa, sillä aiemmin määritellyillä lähtökohdilla peli hyvin todennäköisesti antaisi tälle profiilille juuri mitä tämä haluaa.

3.4.3 Kilpailu

Kuten aiemmassa osiossa kävi ilmi, Abducted tehtiin sokkona ilman mitään ajatusta siitä, kenelle peli on tarkoitettu. Sama pätee myös kilpailuun. Kohderyhmästä poiketen on kuitenkin turhaa edes pohtia Abductedin kilpailijoita, peli kun on vaihtumassa yksinpelistä moninpeliin. Sen sijaan on tärkeää, että jatko-osan mahdollisia kilpailijoita mietitään.

Kun ideana on tehdä kukkulan kuningas moninpeli, jota pelataan selaimesta, oikein montaa siihen painottuvaa peliä ei tule mieleen. Hyvin monesti kukkulan kuningas on jonkun pelin moninpelin erillinen pelimuoto ja siitäkin on voitu tehdä omia variaatioita. Yhdet harvoista peleistä, jotka painottuvat täysin tähän pelimuotoon, ovat Agar.io ja Slither.io. Näissä pelaajan tavoite on kasvattaa itseään muita pelaajia syömällä ja kohota asteikolla korkeimmalle. Oleellinen ero Abductedin jatkoosaan olisi tietenkin pienemmän mittakaavan moninpeli ja se, että peli jaottuu kierroksiin, jossa on voittaja ja häviäjiä, siinä missä .io-peleissä voittaminen koostuu lähinnä siitä ajasta minkä pystyy huipulla pysymään. Lisäksi peleissä eliminoidaan muut pelaajat suoraan, siinä missä Abductedin jatko-osassa tämä tapahtuu epäsuorasti.

Toinen peli minkä voisi luokitella Abductedin jatko-osan lähimaastoon olisi Town of Salem. Tässä pelissä on tarkoitus tehdä suhteita muiden pelaajien kanssa ja huijata näitä, selvitäkseen voittajana. Peli onkin hyvin strategia painotteinen ja sisältää useita tavoitteita ja rooleja, mutta perusidea on hyvin lähellä: selviä viimeisenä hengissä. Toimintalajityypissä pelit eivät kuitenkaan kohtaa ollenkaan.

Mikäli katsoo Abductedin jatko-osaa korkeammalla tasolla, kilpailua löytyy enemmän, mutta taso on kovempaa. Noin esimerkkeinä Splatoon, Overwatch, Counter Strike: Global Offensive. Nämä ovat kuitenkin erittäin isoja pelejä isoilta peliyrityksiltä, joten niitä ei ole mitään järkeä luokitella pelin kilpailuksi. Tämä siksi, koska vaatii hyvin paljon työtä päästä mainittujen pelien tasolle, mikä on taas tämän työn kannalta epärealistinen tavoite.

Onko Abductedin jatko-osan kilpailijoissa jotain, mitä voisi ottaa suunnittelussa huomioon? Agar.io ja Slither.io pelien ollessa melkein ainoat oleelliset kilpailijat, on oikeastaan hyvin vähän millä Abductedin jatko-osa voi kilpailla näitä vastaan tehokkaasti sellaisenaan. Agar.io ja Slither.io peleillä on kuitenkin se etu, että pelin valtavien kokojen vuoksi näitä voi pelata hyvin nopeasti myös yksin ja missä vain. Abductedin jatko-osa taas väistämättä vaatisi muita pelaajia, sillä tekoäly-vastustajat tuskin antavat yhtä suurta nautintoa. Pelaajakunnan koko voisi myös osoittautua ongelmaksi pelaajien kohdalla, jotka eivät järjestä pelejä kaverien kanssa.

Vaan kuten mainittua aiemmin, pelien välillä on yksi merkittävä ero: siinä missä Agar.io ja Slither.io peleissä muut pelaajat eliminoidaan itse suoraan (taikka muiden toimesta), ei Abductedin jatko-osassa suora eliminointi ole mahdollista, mikäli pelaaja ei pelaa ufoa. Tämä voisikin olla alue johon panostaa suunnittelussa: mitä eri keinoja pelaajien on mahdollista käyttää epäsuorasti eliminoidakseen toisiaan?

Muuten on oikeastaan todettava, että Abductedin jatko-osalla on aivan turha lähteä hyppimään Agar.io ja Slither.io pelien varpaille. Kilpailun epämääräisyydestä sekä sen valtavista kokoeroista johtuen suunnittelussa ei oteta kilpailua huomioon merkittävällä tavalla.

3.4.4 Yhteenveto

Palautteen sekä vähän myös takaisinmallinnuksen pohjalta saatiin ideoita ja ajatuksia, mitä ensimmäiseen peliin tulisi korjata ja mitä tehdä lisää. Pelin koodin laatuun sekä ulkoasuun pitää panostamaan enemmän ja suunnittelussa pitää pyrkiä miettimään, miten kokemusta voisi rikastaa. Eri-tyisesti koodin puolella periytymistä ja optimointia tulisi harrastaa enemmän. Ulkoasun suhteen on löydettävä yhtenäinen tyyli.

Palautteessa toivottiin moninpeliä, ja koska tämä sopii pelin ideaan ja muut ehdotukset pelin mekaniikkoihin tukevat tätä ajatusta, tämä tulee olemaan pääasiallinen lähtökohta jatko-osan suunnitteluun. Moninpelin lisäksi peliin tulisi myös kartalta poimittavia työkaluja, joilla peliä piristää. Moninpelillä onkin tarkoitus toteuttaa aiemmin mainittu kokemuksen rikastaminen.

Kilpailua tarkkaillessa todettiin kilpailun olevan erittäin kovaa ja vaikeaa, sillä pienimmillään kilpailijoilla/vastaavilla peleillä on valtava suosio. Tästä huolimatta todettiin, että suunnittelun kannalta olisi hyvä keskittyä pelin uniikkiin ominaisuuteen: pelaajien eliminointiin epäsuorasti. Tämä tarkoittaa siis työkalujen ja kenttien ideointiin panostamista.

Kohderyhmä pelin kohdalla määriteltiin uusiksi, silläkin uhalla, että joku ensimmäisen pelin kohderyhmästä ei pitäisi moninpeli painotteisesta suunnasta. Tästä ei koeta olevan merkittävää haittaa, sillä ensimmäisellä pelillä ei ole varsinaista fanikuntaa ja uuden suunnan voi myös väittää olevan objektiivisesti parempi pelin idealle.

Näin ollen lähtökohdat Abductedin jatko-osan suhteen on selvitetty. Seuraavassa osiossa aloitetaan varsinainen suunnittelutyö.

4 JATKO-OSAN SUUNNITTELU

Tässä osiossa on tarkoitus raportoida työn ensimmäisestä käytännön osuudesta, jatko-osan suunnittelusta. Varsinaisen suunnittelun aikana pidettiin päiväkirjaa, johon päivittäin tehdyt asiat ja ongelmat kirjattiin muistiin. Raportti on tehty tämän päiväkirjan pohjalta. Lopputuloksena suunnittelun jälkeen syntyi pelisuunnitteludokumentti. Tämän pohjalta tehdään toinen käytännön osuus, eli prototyyppi.

Seuraava teksti on jaettu muutamaan osaan: ensiksi itse pelisuunnitteludokumentista ja sen rakenteesta, sitten esisuunnittelutyöstä ja tämän jälkeen pelin eri osa-alueiden suunnittelusta erikseen. Osa-alueiden tekstit perustuvat lopulliseen suunnitelmaan.

4.1 Pelisuunnitteludokumentti ja sen rakenne

Pelisuunnitteludokumentilla tarkoitetaan asiakirjaa, jossa käydään perusteellisesti läpi mitä itse videopeli tulee käsittämään. Asiakirjaa voi siis pitää eräänlaisena keittokirjana tuotantoa varten. Tzvi Freemanin mukaan (1997, viitattu 29.6.2016) suunnitteludokumentti kuvaa koko projektin ruumiin ja sielun kaikkine yksityiskohtineen ja toteutus tapoineen. Suunnitteludokumentteja voidaan hyödyntää oikeastaan missä vain, mutta tuolla termillä niitä kohtaa lähinnä ohjelmistokehityksen parissa.

Dokumentaation merkitystä ei voi vähätellä pelialallaakaan, sillä kuten Alex Sayenko on kirjoittanut, pelisuunnitteludokumentin teko auttaa tiimin suunnittelijaa ymmärtämään pelin olemuksen ja sekä myös viestimään visionsa muulle tiimille. Dokumentti voi myös auttaa löytämään heikkouksia tai puuttuvia osa-alueita, mitä pelistä vielä puuttuu. (2015, viitattu 29.6.2016). Tämän omistaminen voi myös nopeuttaa työskentelyä, sillä ennakoivalla suunnittelulla voidaan vähentää turhaa työtä ja myös selvittää ongelmakohtia etukäteen. Pään sisälle jätetyt suunnitelmat voivat myös lähteä elämään omaa elämäänsä ja jopa unohtua: dokumentoidut suunnitelmat ovat käytännössä kiveen hakattuja.

Tälle työlle pelisuunnitteludokumentin teko on koko opinnäytteen päätehtävä ja siksin se on hyvin merkittävässä osassa tätä työtä. Tämän lisäksi koko peliä ei ole tarkoitus tehdä prototyyppiä pidemmälle, mutta suunnitteludokumentaation kera se on mahdollista jälkeenkäin.

Työ alkoi pohtimalla, minkälaista rakennetta dokumentille halutaan. Dokumentti oli tärkeysjärjestyksessä ensimmäisenä, sillä tällöin saadaan nopeasti pohja varsinaiselle pelisuunnitteludokumentille. Kun tämä on valmiina, ei tarvitse tuhata aikaa kirjoittaessa ajatuksia muualle, vaan ne voidaan tehdä suoraan suunnitteludokumenttiin. Lisäksi rakenteen miettiminen auttaa myös hahmottamaan itse suunnittelua, eli mitä pitää oikeastaan suunnitella.

Dokumentaatio päätettiin tehdä niin, että ensin määritellään osiot ja näitä osioita aletaan laajentaa yksityiskohtiin. Tämä johtuu kirjoittajan mieltymyksestä, sillä kuten myöhemmin käy ilmi, päinvas-tainen ei toiminut. Muut lähestymistavat ovat myös vaikeita sen suhteen, miten kokonaiskuva pro-jektista syntyy.

Osiot päätyivät koostumaan seuraavista asioista, seuraavassa järjestyksessä:

- Esittely
- Peli yleisesti
- Pelattavuus
- Käyttöliittymä
- Kenttäsuunnittelu
- Grafiikka
- Äänet

Nämä ovat oleellisia ydinasioita pelille, joten nämä kuuluvat olla pelin kuin pelin suunnitteludoku-mentaatiossa. Kahden ensimmäisen osion roolina on tehdä pelin olemus selväksi ja esittely oike-astaan tiivistää koko dokumentin. Tämä ratkaisu siksi, että henkilö joka lukee dokumenttia aivan ensimmäistä kertaa, ymmärtäisi pelin olemuksen mahdollisimman nopeasti. Erityisesti uudelle hen-kilölle kehitystiimissä pelin olemuksen ymmärtäminen nopeasti on hyvin tärkeää, sillä muuten tällä ei ole asiaa tuotantoon. Mainittakoon tässä välissä, että liiketoimintaa ei tässä työssä ollut tarkoitus käydä läpi, joten siihen liittyvät asiat loistavat siksi poissaolollaan dokumentin rakenteista. Pelistä riippuen tämä on tietenkin aihealue, jota ei ole edes pakko lisätä.

Järjestys perustuu lukemiseen: on ensin ymmärrettävä idea, jotta ymmärtää kaiken muun. Jotta ymmärtää käyttöliittymän tai kenttäsuunnittelun, on taas ymmärrettävä mekaniikat. Tarkoituksena oli tehdä luurankomainen rakenne: ensin oleelliset asiat joita ilman peliä ei voi toteuttaa, joita seuraava taas hienommat yksityiskohdat.

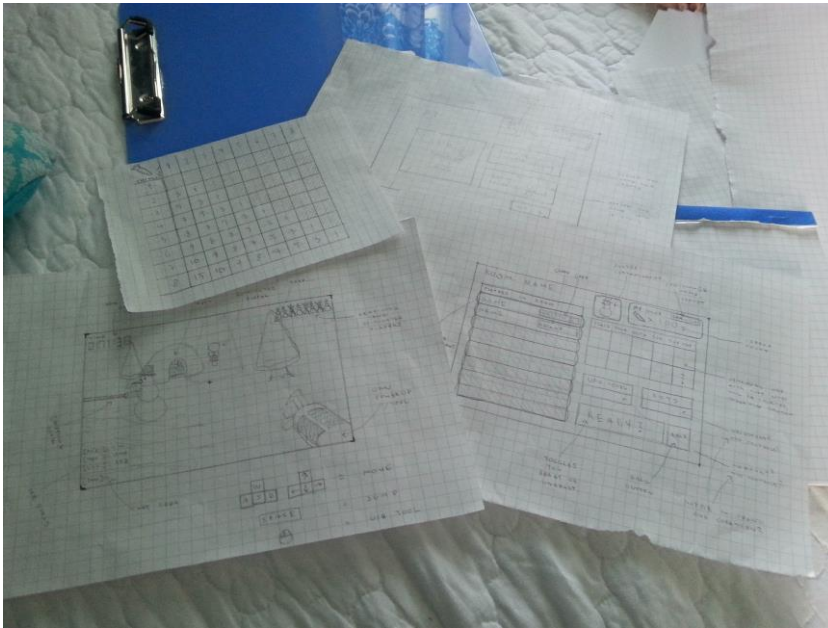
Rakenteen määrittelyn jälkeen osioita tuli pilkkottua vielä pienempiin osiin. Edellä mainitut asiat ovat nimittäin isoja alueita sellaisinaan ja tekstiä olisi vaikea lukea. Näiden sisältöä käydään erikseen niiden omissa osioissa alempana läpi. Tämän lisäksi joka osion otsikon jälkeen lisättiin pieni apu-teksti, mikä sisältää vähän ohjeita siitä, mitä osiossa halutaan saada selville. Tämän tarkoitus olisi auttaa varsinaista suunnittelutyötä. Näiden ohjetekstien avulla olisi jokin ajatus siitä, mitä pitäisi miettiä kunkin osion suhteen. Ohjetekstit syntyivät kirjoittajan aiempien kokemusten pohjalta ja ne pyrittiin tekemään ulkopuolisesta näkökulmasta. Kirjoittamiseen syntyi näin tehtäväluonteinen rytmi. Myöhemmin nämä otettiin pois siisteyden vuoksi.

Tässä vaiheessa tuli tehtyä myös päätös dokumentin kielestä. Kirjoittajalle alan englanninkielinen termistö on tutumpaa ja mikäli haluaa jättää ovet auki kansainväliselle avulle, päätettiin pelisuunnitteludokumentti kirjoittaa englannin kielellä. Tämän jälkeen varsinainen kirjoitustyö alkoi.

4.2 Aivorihi

Pelisuunnitteludokumentin kirjoittaminen suoraan rakenteen määrittelyn jälkeen ei ollut fiksu ajatus. Kävi nimittäin ilmi, että vaikka pelin idea oli selvillä ja taustatyötä sille oli tehty, tämä ei riittänyt hahmottamaan ideaa kunnolla. Oltiin siis tilanteessa, että haluttiin dokumentoida yksityiskohtia, vaikkei kokonaiskuva vielä tiedetty. Tämän ja kirjoittajan päinvastaisten tottumuksien vuoksi oli siis oikeastaan mahdotonta yrittää alkaa dokumentoida suunnitelmia pelin suhteen, kun hyvin paljon pelistä oli täysin pimennossa. Työssä seuraavaksi ei loppujen lopuksi ollutkaan kirjoittaminen, vaan aivorihi.

Aivoriihellä tarkoitetaan tässä yhteydessä varsinaiseksi suunnitteluprosessiksi, siinä missä pelisuunnitteludokumentin teko on käytännössä vain suunnittelun dokumentointia ja toteutuksen suunnittelua. Tässä vaiheessa työtä suunnittelu tapahtui siis enemmän pään sisällä ja kynällä piirrellen. Tämä menetelmä perustuu kirjoittajan aiempiin kokemuksiin. Tarkoituksena on piirtää pelin näkymiä ja pohtia sitä kautta mitä peliin haluaa, miten pelaaja viestii pelin kanssa ja miltä peli näyttää.



Kuvio 10 Aivoriin tuloksia

Aivoriin aikana saatiin aikaan muutama arkki piirustuksia, jotka sisältävät kuvaa siitä, miltä pelin olisi tarkoitus näyttää. Mainittakoon, että kuviossa 10 näkyvät paperit tulivat muuttumaan useampaan otteeseen.

Pelin ymmärtämisen takia aivoriin aikana tavoitteeksi muodostui vastata seuraaviin kysymyksiin: miltä peli näyttää, mitä pelaajan on nähtävä ja mitä pelaajan on pystyttävä tekemään? Nämä ovat kuitenkin kysymyksiä, joihin on tuotantoa varten tiedettävä vastauksia. Kaksi ensimmäistä kysymystä muodostivat idean sille, että joka paperille piirrettäisiin pelin näkymä niin kuin se pelaajalle näkyisi. Näkymien sisällöt taas syntyivät kolmannesta kysymyksestä.

Näkymien suunnittelun ajatusmalli oli seuraava: koska peli on moninpeli, on jossain ruudussa oltava nappi, josta pääsee liittymään peliin. Ja koska sellainen nappi on olemassa, on oltava olemassa näkymä, jossa näkee muut pelaajat ja joka sisältää napin jolla poistua aiempaan näkymään. Tällä ajatusmallilla pelin käyttöliittymä ja mekaniikat alkoivat lopulta hahmottua. Ensimmäisenä ajatuksena oli sisällyttää tarvittavat mekaniikat ja ominaisuudet eri näkymiin, jonka jälkeen tuli vasta ajatus asettelusta. Mekaniikat koostuivat moninpeliominaisuuksista, sekä lähtökohtien aikana pohdituista työkaluista ja voimista. Myöhemmin suunnittelun aikana tuli myös ajatus pelaajan palkitsemisesta porkkanoilla, joilla voisi ostaa koristeita omalle hahmolleen. Mekaniikkoihin lukeutui siis

myös porkkanoiden kerääminen ja hahmon muokkaus. Tämä idea perustui nykyiseen trendiin moninpeleissä, jossa itsensä voi erotella muista luomalla omanlaisensa hahmon. Tämän lisäksi pelistä piti löytyä myös asetuksia ja apua niille ketkä sitä tarvitsevat. Näiden pohjalta pelin näkymät syntyivät ja suunnittelua helpotti se, että mekaniikat ja ominaisuudet liittyivät toisiinsa.

Asettelyn suhteen ajatuksena oli pitää asiat yhtenäisesti aseteltuina, sillä yhtenäinen käyttöliittymä näyttää paremmalta ja mahdollistaa pelaajalle myös mukavia rutiineja pelin näkymiä navigoidessa. Muuten tarvittavat painikkeet ja kuvat määrittivät pitkälti sen, missä ne sijaitsivat ja usein tämä oli siitäkin huolimatta tapauskohtaista. Tarkoitus oli esimerkiksi pitää paluu-nappi aina samassa kohdassa, sillä sen vaihteleva sijainti ei koskaan muodostaisi rutiinia pelaajan kanssa ja muuttuisi vain ärsykeeksi. Pelaajan asetuksiin liittyvät asiat oli taas tarkoitus pitää samassa kohdassa myös eri näkymissä, aivan samasta syystä.

Aivoriihi jatkui, kunnes pelin ideasta oli syntynyt niin selkeä kuva, että olisi mahdollista alkaa dokumentoida ideoita. Kaikkia mahdollisia näkymiä pelistä ei tässä vaiheessa vielä piirretty ja itse suunnitteludokumentin kirjoittamisen aikanakin eri näkymiä uusittiin useampaankin otteeseen puuttuvien näkymien piirtämisen lisäksi. Syynä tähän oli yleensä uusi mekaniikka tai mekaniikan poistuminen sekä myös tyylilliset seikat. Erityisesti näkymä, jossa pelaajat näkisivät muut pelaajat listana, koki paljon muutoksia. Tässä näkymässä piti nimittäin olla paljon asiaa, joten elementtien asettelua oli vaikea saada tarpeeksi selkeäksi. Tällöin tämä näytti hyvin ruuhkaiselta ja sisälsi myös ominaisuuksia, jotka hyödyn sijaan aiheuttaisivat hämmennystä. Lopulta eräs mekaniikoista otettiin pois, joten näkymä oli piirrettävä uudelleen ja tämä sujui paljon paremmin. Muussa tapauksessa näkymään sijoiteltuja elementtejä olisi pitänyt pienentää tai jopa sivuttaa.

Lopputuloksena aivoriihen aikana syntyi lukuisia piirustuksia ja selkeämpi kuva pelin rakenteesta. Tekstiä aivoriihen aikana ei syntynyt oikein muussa muodossa, kuin selittävinä teksteinä näkymäpapereissa.

4.3 Pelin yleiset asiat

Kun aivoriihi oli tehty, oli vuorossa varsinainen dokumentointi. Dokumentointi alkoi suunnitellun rakenteen ensimmäisistä osista, joiden tarkoituksena oli selkeyttää pelin idea. Nämä olisivat paitsi helpompia kirjoittaa pois työjärjestyksestä, myös auttaisivat hahmottamaan pelin ideaa kirjoittajalle

entisestään. Alun perin pelin ideoita selkeyttäviä osia oli vain yksi: peli yleisesti -osio. Myöhemmin heräsi kuitenkin ajatus siitä, että tiiviimpi esittely-osio voisi auttaa dokumentin lukemista ja pelin idean ymmärtämistä.

Kirjoittaminen alkoi kuitenkin tästä niin sanotusta yleis-osiosta, mikä jaettiin selkeyden vuoksi pariin muuhun osuuteen:

- Lajityyppi
- Kohdealusta
- Ajatusmaailma
- Pelin virta

Näitä osioita edelsi lyhyt hissiesitelmäksikin kutsuttu teksti. Tällä tarkoitetaan siis esittelyä, joka suullisesti kestää korkeintaan minuutin ja tekstinä on parhaimmillaan vain pari lausetta. Pelin idea oli siis kukkulan kuningas -pelimuotoon keskittyvä verkkopeli, jossa pelaajat ovat lumiukkoja ja näitä jahtaa tekoälyllä toimiva ufo. Pelin idean ollessa hyvin yksinkertainen, idean tiivistäminen pariin lauseeseen ei tuottanut ongelmia.

Lajityypissä oli tarkoitus kertoa vastaavista peleistä ja ilmaista pelin eroavaisuus näistä kilpailijoista. Teksti koostui käytännössä samoista asioista, mitä aiemmin lähtökohtia selvittäessä saatiin selville. Ongelmia ei ilmentynyt, mutta osio oli vaikeampi kirjoittaa, kun tavoitteena pelillä ei ole tehdä rahaa. Pelin lopulliseksi lajityypiksi määriteltiin ensimmäisen persoonan toimintapeli verkossa.

Kohdealustassa oli tarkoitus kertoa ja perustella, millä laitteella peliä pelataan. Tämäkin ajatus oli jo selkeä lähtökohtia selvitetessä: peli tulisi tietokoneelle ja olisi pelattavissa selaimesta. Tämä siksi, että peli olisi niin pieni ja yksinkertainen, että julkaisu erillisenä ohjelmana tuntui tarpeettomalta. Lisäksi Game Jolt -sivusto tekisi julkaisun tässä muodossa helpoksi ja koska peli liitetään kyseiseen sivustoon, käy myös järkeen hyödyntää sivuston tarjoamia ominaisuuksia, kuten pilvitalennuksia. Tässä välissä oli myös ajatuksena, että muita alustoja ei mietitä ollenkaan: konsoleille peliä ei voi kehittää ilmaiseksi ja mobiililla ensimmäisen persoonan pelit eivät toimi yhtä hyvin muilla alustoilla. Mobiililla tämä johtuu kontrollien heikommasta hallittavuudesta pelityypin parissa. Tekstissä mainittiin myös pelimoottori, jolla peli tehtäisiin. Unity 3D -ohjelmisto kuitenkin mainitaan esimerkiksi mekaniikkoja esitellessä hyvin usein, joten käy järkeen, että tämä tehdään selväksi aikaisin.

Ajatusmaailmassa tarkoitus oli selkeyttää pelaajan tunnetilat pelin aikana. Tämä on tärkeää tietoa siksi, että tällöin on selkeämpää, mitä pelillä halutaan tehdä. Jos pelin on tarkoitus olla hauska, pelaajan tulisi huvittua ja jopa nauraa. Tällöin suunnittelussa on otettava huomioon, millä tämä tavoite saavutetaan. Tässä pelissä ajatusmaailmana on hämmennys ja huumori: lumiukkoja jah-taava ufo on kuitenkin hämmentävä asetelma ja myös surullinen, kun ajattelee että voittaminen tapahtuu muut uhraamalla. Pelin ulkoasu ja moninpeliominaisuudet sen sijaan tuovat tähän kontrastin antamalla päinvastaisia tunteita, mistä taas voi syntyä huumoria. Tähän ei ollut suunnittelu-mielessä sen enempää ajatusta kuin selittää, mitä ajatuksia idea sellaisenaan synnyttää pelaajassa. Voisi siis sanoa, että pelin idea tuli tässä tapauksessa tavoitteita edelle.

Pelin virran (engl. Game flow) tarkoituksena oli selkeyttää peliä ja sen rakennetta entisestään kaavioiden kautta. Kaaviot siksi, koska kuvilla pelin rakennetta olisi helpompi kuvailla tekstin sijaan. Kaavioita tehtiin lopulta vain kaksi: ydinsilmukka (engl. Core loop) ja pelivirtataulu (engl. Game flow board). Ydinsilmukka tiivistää hyvin korkealla tasolla mitä pelissä tehdään ja vain parilla laatikolla tekstiä. Pelivirtataulu taas laajentaa ydinsilmukkaa ja antaa muutamia yksityiskohtia. Kaavioita tehdessä pohjana toimi netistä löydetty mallipohjat, joita muokattiin tarpeen mukaan.

Myöhemmin tehdyssä esittely-osiossa tavoitteena oli tiivistää kaikki pelin oleellinen tieto yhteen sivuun: pelin idea, mekaniikat, kohdeyleisö, kohdelaitteisto sekä ominaisuudet, jotka erottavat pelin kilpailijoista. Näin kuka tahansa voisi vilkaista vain yhtä sivua koko dokumentista ja ymmärtää mistä ideassa on kyse. Muu tieto ei ollut tarpeellista ja edellä mainitut asiat riittävät selventämään pelin luonteen. Esittelyn rakenne pohjautui The Legend of Johnny -sivustolla esitettyyn high concept document pohjaan. Osiota kirjoittaessa ei esiintynyt ongelmia tai massiivista tarvetta korjata tekstiä taikka rakennetta. Pelin idea oli kuitenkin hyvin selkeä aivoriihen jälkeen.

Yhteenvetona pelin yleiset asiat olivat käytännössä jo ennalta määriteltyä tietoa. Pelisuunnittelu-dokumentin kannalta helpoin osuus, sillä varsinaisesti suunnittelulle ja ajatustyölle ei ollut suurta tarvetta. Paljon asioita oli kuitenkin jo selvennetty aiemmin lähtökohtia selvitetäessä. Myös kaaviot olivat helppoja tehdä, sillä näiden teosta on aiempaa kokemusta ja hyviä mallipohjia löytyy hyvin paljon.

4.4 Pelattavuus

Seuraavana työssä oli vuorossa pelattavuuden dokumentointi. Aivoriihen, lähtökohtien selvittelyn ja pelin yleisten asioiden selvittelyn lopputuloksena pelistä oli tässä vaiheessa hyvin selkeä kuva: nyt yksityiskohdat olisi vain laitettava paperille ja pelattavuus oli tärkeysjärjestyksessä ylimpänä.

Tämä osuus jakautui useampiin osiin selkeyden vuoksi. Näitä osia taas edelsi yleinen kuvaus siitä, miten peli toimii ja miten sitä pelataan. Kuvaus pohjautui aivoriihessä tehtyihin suunnitelmiin ja sen tarkoitus oli selkeyttää pelin toiminnallisuutta ennen tarkkojen yksityiskohtien selvittelyä, mikä tapahtuisi aiemmin mainituissa osioissa.

Kyseiset osiot ovat:

- Mekaniikat
- Pelaajan tiedot
- Asetukset
- Tallennus

Mekaniikoissa oli tarkoitus käydä läpi kaikki pelin mekaniikat ja selittää miten ne voitaisiin toteuttaa. Mekaniikoilla tarkoitetaan ominaisuuksia ja rajoja pelissä, millä pelaaja kommunikoi pelin kanssa. Niillä myös tarkoitetaan sääntöjä, joilla peli etenee. Osiossa siis käydään läpi se, miten peli oikein toimii. Koska kyseessä on suunnitteludokumentti, myös tapa jolla mekaniikka toteutetaan, on oleellista tietoa sisällyttää.

Toteutussuunnitelmissa oli tavoitteena antaa vain vähän suuntaa sille, miten toteutus voisi tapahtua, sillä muuten työ paisuisi tarpeettomasti. Varsinaisen kehityksen aikana kuitenkin hyvin usein selviää, ettei joku yksityiskohtainen suunnitelma toimikaan, jolloin suunnitelman teko hyvin pienissä yksityiskohdissa muodostuu ajanhukaksi. Tämä taas johtaa siihen, että suunnitteludokumenttia on päivitettävä selkeyden vuoksi ja tämä herkästi laiminlyödään kehityksen aikana. Lopputulokseksi jää siis dokumentti, joka ei ole ajan tasalla, mutta minkä on tarkoitus selventää pelin olemus myös uusille kehittäjille.

Loppujen lopuksi toteutussuunnitelmien syvyys vaihteli suuresti. Esimerkiksi pelin moninpeliominaisuudet jäivät kohtalaisen pintapuolisiksi, mutta toisaalta taas pelissä pelaajia jahtaavan ufon

tekoäly sai hyvin tarkkaa tietoa sen toiminnallisuudesta. Tämä johtui kirjoittajan kokemuksesta Unity 3D -ohjelmiston teknologian suhteen: osa asioista on ennalta tuttuja, osa taas ei.

Laajuutensa vuoksi Mekaniikat-osio oli pakko jakaa vielä pienempiin osiin:

- Moninpeli
- Voimat ja työkalut
- Porkkanat
- Hahmomuokkaus
- Ufon tekoäly

Nämä osiot tulivat siitä ajatuksesta, mitä peli oli tarkoitus sisältää. Pelin idea oli jo hyvin selvillä lähtökohtien ja aivoriihen myötä, joten osiot olivat lähinnä vain ajatusten dokumentointia. Tavoitteena oli pitää osioissa sellainen rytmi, että ensin mekaniikka selitetään korkealla tasolla, mitä seuraisi taas yksityiskohtainen toteutussuunnitelma.

Moninpelin suhteen päätökset tulivat pitkälti aivoriihen aikana tehdyistä käyttöliittymäkuvista. Näissä esiintyi selkeitä näkymiä, joista pystyi näkemään mitä pelaajan tulisi pystyä tekemään. Aivoriihen aikana päätökset ja ideat sen sijaan tulivat kirjoittajan kokemusten pohjalta verkkopelien kanssa. Paljolti pelin verkko-ominaisuuksien järjestelyjä inspiroi Nintendon tuottama Splatoon-verkkomoninpeli. Tyypillisesti indie-moninpeleissä tarjotaan mahdollisuutta tehdä niin sanottuja huoneita, joita voidaan nimetä, listata ja joihin liittyä. Tämä tapahtuu indie-moninpeleissä siksi, koska se on teknisesti hyvin helppoa tehdä. Splatoonin tapa olisi kuitenkin järkevämpi: vaikka huoneita on olemassa, erillisiä huoneita ei pysty katselemaan vaan pelaaja heitetään heti seuraavaan mahdolliseen huoneeseen. Tällä estetään se, että kaikki pelaajat vain tekevät omia huoneitansa ja odottavat turhaan muita pelaajia. Tämä on toki pelaajan itsensä vastuulla, mutta toisaalta tämä taas on hyvin turha mutka käyttöliittymässä.

Suunnitelmiin sisältyi tietenkin myös tarjota Splatoonin lailla yksityisiä otteluita: onhan pelaajien pystyttävä pelaamaan myös vain kaveripiirin kesken. Tämän toteutus oli vähän vaikeampi suunnitella, sillä kaveri-järjestelmä näin pienelle pelille on vähän liikaa, eikä Game Jolt tarjoa tälle muutenkaan helppoja ominaisuuksia. Kaveri-järjestelmässä pystyisi solmimaan ystävyysuhteita muiden käyttäjien kanssa ja pelissä pystyisi näkemään kavereistaan listan. Tyypillisesti moninpeleissä tätä listaa voisi käyttää muodostamaan otteluita. Loppujen lopuksi toteutus oli seuraava: anna pelaajan luoda huone, jonne joku toinen voi liittyä vain, jos tietää antaa kyseisen huoneen nimen.

Nimi toimii siis tavallaan salasanana, eikä huoneita listata missään. Teknisesti tämä olisi mahdollinen toteuttaa ja mahdollistaa yksityiset ottelut helposti. Myöhemmin kävi kuitenkin ilmi, että satunnainen liittyminen ei erottelisi yksityisiä otteluita haussa. Onneksi huoneille kuitenkin on mahdollista antaa yksilöllisiä arvoja ja hakua on mahdollista rajata näiden perusteella.

Kun pelaaja liittyy huoneeseen, tämä näkisi näkymän jossa muut huoneen pelaajat on listattu. Tämän toteutus poikkesi vähän Splatoonista. Kyseisessä pelissä peli alkaa heti, kun huone on täynnä 8 pelaajaa ja vain kun tämä ehto on täyttynyt. Tässä pelissä tämä ei kuitenkaan toimisi, sillä ottaen huomioon pelin koon, ei voi odottaa, että 8 pelaajaa on saatavilla hetkellä minä hyvänsä. Automaattisuuden sijaan manuaalinen ottelun aloitus toimisi siis parempi. Ajatuksena oli antaa pelaajille kaksi tilaa: valmis ja valmistelee. Pelaaja voisi hallita tätä tilaa ja kun huoneessa on enemmän kuin yksi pelaaja, joiden tilana on valmis, peli alkaisi. Tälle ei varsinaisesti ollut mitään inspiraation lähdettä, mutta tällä pystyisi toteuttamaan manuaalisen ottelun aloituksen. Splatoonin tyyliin taustalla kuitenkin pyörisi laskuri, joka asettaisi pelaajan tilaan valmis automaattisesti. Tämä siksi, että välitetään tilanne, jossa joku pelaaja on liittynyt peliin, mutta syystä tai toisesta ei koskaan aseta itseään valmiiksi. Mikäli tätä ei olisi, ottelu ei voisi koskaan alkaa ja muut saisivat vain odottaa turhaan. Vastaava laskuri suunniteltiin myös ottelun tuloksien näyttämisen ajaksi, samasta syystä.

Alun perin moninpelissä oli tarkoitus antaa mahdollisuus äänestää ottelun karttaa, hallita jahtaavan Ufon aggressiivisuutta sekä ottaa otteluun mukaan tekoäly-pelaajia. Nämä tekivät kuitenkin käyttöliittymästä hyvin ruuhkaisen ja etenkin tekoäly-pelaajat olisivat iso työ tehdä kunnolla. Lisäksi jos otteluita erottelisi Ufon ominaisuuksien ja tekoäly-pelaajien perusteella, pilkkoiisi tämä pelaajakuntaa tarpeettomasti pienempiin osiin. Nämä poistettiin suunnitelmista melkein välittömästi.

Voimat ja työkalut -osio koostui käytännössä isosta listasta asioita, joilla pelaajat voisivat paremmin paeta Ufolta tai vaikeuttaa muiden pelaajien peliä. Listojen lisäksi osioon tuli myös yleisiä asioita, kuten voimien ja työkalujen poimiminen, käyttö ja esiintyminen ympäristössä. Ideoinnissa teemana oli juurikin itsensä voimistaminen ja muiden heikentäminen. Tavoitteena oli keksiä mahdollisimman tasapainoisia voimia, joilla pystyisi epäsuorasti vastustajien peliä heikentämään. Osa ideoista tuli jo oikeastaan lähtökohtien selvittelyn aikana ilmenneistä ideoista, kuten heittokoukku. Tärkeää ideoissa oli myös realismi: pystyykö tätä toteuttamaan, erityisesti moninpeliympäristössä? Toteutussuunnitelmat olivatkin loppujen lopuksi melko yksityiskohtaisia. Varsinainen lista voimista ja työkaluista tehtiin taulukkomuotoon selkeyden vuoksi, johon tehtiin myöhemmin myös pientä konseptitaidetta sen suhteen, miltä voimat ja työkalut ympäristössä näyttäivät.

Porkkanat-osiossa käytiin läpi mitä porkkanoilla voi tehdä ja mistä niitä saa. Kuten aiemmin mainittua, porkkanat olivat idea liittyen pelaajan hahmon muokkaukseen. Ensimmäinen ajatus porkkanoiden keräämisestä oli otteluiden sijoitukset. Mitä korkeammalle sijoitukselle pääsee, sitä enemmän porkkanoita saa. Tasapainon vuoksi tämä määrä olisi suhteutettuna pelaajien määrään ja reiluuden vuoksi kesken ottelun lähteneet pelaajat eivät saisi mitään, eivätkä nämä vaikuttaisi muiden pelaajien porkkanamääriin. Toisin sanoen vain lähtökohtaisella porkkanamäärällä on merkitystä. Dokumentaatio sisälsi lopulta taulukon, josta voi suoraan katsoa monta porkkanaa saa, jos sijoittuu vaikka toiseksi viiden pelaajan ottelussa. Tämän hahmotusta selventämään taulukon jälkeen oli myös pari käytännön esimerkkiä. Varsinaiset määrät taulukossa olivat aluksi hyvin pieniä, mutta näitä nostettiin myöhemmin tavoitteena tasapainottaa koriste-esineiden hinnat ja saadut porkkanat. Tavoitteena oli myös tehdä ostamisen kynnyksestä mahdollisimman alhainen, että pelaajan ei tarvitse pelata kymmeniä otteluita vain saadaakseen yhden koristeen.

Hahmomuokkaus-osio kulki tavallaan porkkanat-osion kanssa käsi kädessä, mutta rakenteeltaan muistutti enemmän voimat ja työkalut -osiota. Tämän mekaniikan pointtina olisi luoda pelaajalle palkinto ja jokin otteluiden ulkoinen tekijä, johon pelissä keskittyä. Pelkkä otteluiden pelaaminen voi olla lähinnä tylsää, mutta ulkoinen tekijä voi päinvastoin jopa motivoida pelaamaan. Pelin itsensä tulee toki olla jo riittävän hauska, mutta lisätehtävä ei ole pahitteeksi, jos se tukee muuta peliä. Idea mekaniikalle syntyi yleisestä trendistä moninpeleissä, missä oman hahmon voi muokata mieleisekseen. Tämä käy myös temaattisesti järkeen, sillä yleensä lumiukot koristellaan oman miehen mukaan. Lopullisena ajatuksena oli, että pelaaja voisi päästä tiettyyn näkymään, jossa tämä voi sekä ostaa, että koristella pelihahmonsa. Ideana myös oli, että pelaaja voisi muokata lumiukon rakennetta, mutta vaikeuden vuoksi tämä jätettiin suunnittelupöydälle. Dokumentaatioissa selitettiin miten hahmomuokkaus toimisi ja tämä myös sisälsi alustavia listoja eri koristeille konseptitaiteineen. Tällaisessa pelissä koristeita tulisi olla satoja, mutta jotta suunnittelu ei menisi jäihin tämän työn suhteen, ei listoihin tehty muutamaa vaihtoehtoa enempää koristeita.

Ufon tekoäly -osiossa käytiin läpi, miten ufo käyttäytyy ja toteutussuunnitelma käyttäytymiselle oli lopulta melko yksityiskohtainen. Tekoälyn selventämiseksi dokumentaatioon tuli jopa kaksi pseudokoodin pätkää, millä tarkoitetaan siis koodia joka ei oikeasti ole koodia, mutta mikä kirjoitetaan selkokielellä kuin se olisi koodia.

Pseudokoodia on esimerkiksi:

```
Poimitaan esine
Jos esine on omena
    Se syödään
    Ja roskat laitetaan roskikseen
Muutoin
    Heitetään esine ikkunasta
```

Tämän tarkoituksena oli puhtaasti selkeämpi kuva tekoälyn toiminnasta. Ideat ja vaatimukset tekoälyn suhteen tulivat ensimmäisestä pelistä, jota tutkittiin aiemmin lähtökohtien selvittämisen merkeissä. Tekoälyn toiminnallisuus säilyi oikeastaan samanlaisena, mutta poikkeuksena oli keksiä, miten tekoäly käyttäytyy, kun jahdattavia on useampia. Loppujen lopuksi päädyttiin siihen ratkaisuun, että tekoäly keskittyisi yhteen pelaajaan kerralla ja vaihtelisi kohdettaan satunnaisin väliajoin. Tällöin tekoäly olisi vähän ennalta-arvaamattomampi ja sekoittaisi pelaajia hyvin. On kuitenkin tärkeää tällaiselle kilpailulliselle pelille, että pelaajat eivät ala huomata rutiineja, joita käyttää hyväksi. Tässä on toki riskinä, että tekoäly on liian ennalta-arvaamaton ja täten epäreilu, mutta varasuunnitelmana on toteuttaa kohteen vaihto etäisyyden mukaan.

Tässä vaiheessa pelin päämekaniikat oli käyty läpi. Seuraavaksi oli tarkoitus siirtyä vain järjestyksessä seuraavaan osioon. Aihepiiri ei kuitenkaan muuttuisi vielä pelattavuudesta.

Asetukset-osio tuli sisältämään vain listan ominaisuuksia, joita pelaaja voi muokata mieleisekseen pelissä. Etenkin tietokonepelaajille tämä on erittäin tärkeä ominaisuus. Vaan oli peli mikä hyvänsä, pelaajilla on niin eroavat tottumukset joidenkin mekaniikkojen suhteen, että vaihtoehtoja on hyvä olla olemassa. Ensimmäisen persoonan peleissä etenkin hiiren herkkyys (engl. Mouse sensitivity) ja näkökentän leveys (engl. Field of view) ovat pelaajien kaipaamia asetuksia, sillä tottumukset näiden suhteen vaihtelevat suuresti. Tavoitteena olikin antaa mahdollisimman paljon eri vaihtoehtoja asetuksien suhteen, siitäkin huolimatta, että asetuksesta hyötyy vain pieni osa pelaajakunnasta. Pääasiassa ideat asetuksien suhteen tuli kirjoittajan omista kokemuksista. Dokumentin rakenteen suhteen tehtiin lista, johon selitettiin jokaisen ominaisuuden merkitys, oletusarvoja sekä miten tätä hallitaan itse pelissä.

Pelaajan tiedot -osio tehtiin myöhemmin tallennus-osion yhteydessä ja tämä siksi, että päätettiin tehdä pelaajan tietojen tallennus pilveen. Game Joltin Unity 3D API kuitenkin mahdollistaa Game Joltiin pääsyn itse pelin sisällä ja selainpelien suhteen tätä on myös helpotettu esimerkiksi automaattisella sisäänkirjautumisella. Päätös aiheutti myös sen, että peliä ei voisi pelata ilman, että pelaaja on rekisteröitynyt Game Jolt -sivustolle ja kirjautunut peliin sisään näillä tunnuksilla. Tämä siksi, koska muuten pelaajan tietojen hallinnasta tulisi hankalaa ja huijaaminen olisi helppoa. Tälle on toki vaihtoehtoja, mutta jotta rakenteet pysyvät yksinkertaisina, päätettiin mennä tällä ratkaisulla.

Osio itsessään ei vaatinut juuri ihmeellisempää suunnittelua, osio lähinnä selvensi toteutussuunnitelmaa muille pelin osa-alueille. Tarkoitus olikin tehdä selväksi, mitä tietoa pelaajista pidetään yllä ja missä muodossa.

Tallennus-osio meni käsi kädessä pelaajan tiedot -osion kanssa ja kuten aiemmin mainittua, tämän aikana kävi myös ilmi tarve pelaajan tiedot -osiolle. Alun perin tässä osiossa oli tarkoitus tehdä samat asiat kuin pelaajan tiedot -osiossa, mutta selkeyden vuoksi nämä jaettiin kahtia. Siinä missä pelaajan tiedot -osiossa oli tarkoitus selittää mitä tietoa pelaajasta pidetään yllä, tallennus-osiossa oli tarkoitus selittää, miten tallennus tapahtuu käytännössä. Pelaajan tiedot -osion lailla tämä osio liittyi enemmän toteutussuunnitteluun. Tarkoituksena oli vain selvittää tuotantoa varten, millä tavalla tallennus toteutetaan.

4.5 Käyttöliittymä

Pelattavuuden jälkeen seuraavana työssä oli vuorossa käyttöliittymän suunnittelu. Käyttöliittymällä tässä yhteydessä tarkoitetaan graafisen käyttöliittymän suunnittelun lisäksi myös sitä, miten pelaaja kommunikoi pelin kanssa sekä miten tämä oppii pelin säännöt ja pelin kanssa kommunikoinnin. Tämän vuoksi esimerkiksi pelin kontrollit selitettiin vasta tässä osiossa, vaikka yhteys pelattavuuteen on selkeä.

Selkeyden vuoksi osio jaettiin seuraaviin osiin:

- Näkymät
- Kontrollit
- Apu

Näkymät-osiossa oli tarkoitus esitellä kaikki pelin päänäkymät. Tämä tapahtui piirtämällä ensin kaikki tarvittavat näkymät, mikä aloitettiin jo aivoriihen aikana. Tämän jälkeen kaikki näkymät listattiin ja jokaisen piirretyn kuvan yhteyteen selitettiin mitä pelaaja pystyy tekemään kyseisessä näkymässä. Puuttuvat näkymät suunniteltiin samoilla periaatteilla kuin aivoriihen aikana, eli niin että visuaalisesti näkymät ovat yhtenäisiä ja vastaavat pelaajien tarpeita. Tämä osio tehtiin melko myöhään, sillä näkymiin kohdistui suunnittelun aikana muutoksia.

Kontrollit-osiossa listattiin pelaajalle mahdolliset toiminnot ja tapa, jolla tämä toiminto aktivoidaan. Lista siis sisälsi toimintonäppäin pareja. Näppäimet perustuivat tyypillisiin näppäinasetuksiin muissa ensimmäisen persoonan peleissä. Tämä siksi, että perinteistä poikkeavat näppäimet voivat häiritä peliä ja olla myös vaikeampi omaksua. Perinteiset näppäimet ovat nopeampi ottaa käyttöön jo pelkän tottumuksen kautta, mikä vähentää opettamisen määrää. Kaiken varalta asetusten suunnitelmiin sisällytettiin myös näppäinten vaihto. Yksi asia, mikä suunnittelussa jäi vielä epäselväksi, oli juoksukyvyn tarve. Tämän tarve selviäisi, kun pelistä saadaan prototyyppi testattavaksi.

Apu-osiossa oli tavoitteena selventää, miten pelaaja oppii pelaamaan peliä. Ajatuksena oli, että pelaajaa yritettäisiin opastaa hyvin vähän. Näin siksi, että pelin mekaniikat ja kontrollit ovat kuitenkin hyvin selkeitä, näppäimiä myöten. Pelaajan kädestä pitely näiden suhteen voidaan siis kokea vain ärsykkeeksi. Tästäkin huolimatta osa asioista jää helposti epäselväksi, jolloin syntyi idea tehdä oma näkymä pelaajan avustamiselle. Tässä näkymässä pelaaja voi selailla ohjetekstejä, jossa esimerkiksi voimia ja työkaluja selitetään. Lisäksi syntyi myös idea siitä, että heti pelin alettua joka pelaajalle näytetään teksti, joka kertoisi, että tämän on pyrittävä selviämään viimeisenä. Tällä tavoin ensikertalaiselle syntyy jo heti ajatus siitä, mikä pelissä on tavoite. Loput ilmenevät pelatessa tai vaihtoehtoisesti aiemmin mainittujen ohjetekstien kautta.

4.6 Grafiikka ja ääni

Pelisuunnitteludokumentin rakenteen mukaan seuraavana olisi ollut kenttäsuunnittelu, mutta työn aikana grafiikka ja ääni -osiot tehtiin ensin. Kenttäsuunnittelu vaatisi sen verran ajatustyötä ja tällä ei varsinaisesti olisi merkittävää yhteyttä muihin osuuksiin, joten oli helpompaa ensin tehdä helpommat osiot pois päiväjärjestyksestä.

Rakenteellisesti grafiikka ja ääni -osiot tuli tehtyä samalla tavalla pelisuunnitteludokumenttiin: kumpikin jaettiin kahteen samanlaiseen osioon. Ensimmäinen osio sisälsi tyylimäärittelyt, eli esimerkiksi mitä värejä halutaan grafiikassa käyttää, tai miltä taustamusiikin pitäisi kuulostaa. Tämän tarkoitus oli selkeyttää tyyli, jolloin esimerkiksi artisti ymmärtää minkälaista grafiikkaa tai ääntä peliin tarvitaan. Toinen osio sisälsi taas listan tarvittavista aseteista, minkä tarkoitus on antaa suuntaa siihen, mitä artistin tulisi tehdä peliä varten. Listoihin kului ehkä tarpeettomasti aikaa, mutta toisaalta listojen sisältämien yksityiskohtien takia assetit olisi nyt helppo siirtää tehtäviksi scrum master -sovellukseen.

Kun graafista tyyliä mietittiin, tässä oli tavoitteena vähintäänkin löytää yhtenäinen tyyli. Ensimmäisessä pelissä nimittäin yksi visuaalien isoimmista vioista oli epäyhtenäinen tyyli assettien välillä. Lumiukot, ufo ja maasto olivat yksinkertaisia, siinä missä puut mukailivat realistista tyyliä. Yksinkertaisuuden vuoksi graafiseksi tyyliksi päättyi sarjakuvamainen lowpoly 3D-grafiikka. Lowpolylla tarkoitetaan 3D-grafiikkaa, jossa yksityiskohtia on erittäin vähän. Tämä tyyli olisi helppo toteuttaa, se näyttäisi hyvältä ja se olisi myös laitteiston kannalta kevyt. Tyyli on kyllä hyvin yleisesti käytetty, mutta toisaalta se taas sopii hyvin pelin ajatusmaailmaan. Persoonallisuuden merkitystä ei voi vähätellä pelien graafisessa tyyliässä, mutta tässä tapauksessa tuotannon realismi tulisi tielle hienommissa suunnitelmissa.

Pelin värimaailma perustui asetelmaan: koska pelissä on lumiukkoja, käy järkeen, että ympäristö on talvinen. Tämä siis tarkoittaa kylmiä värejä, pääasiassa harmaan ja sinisen eri sävyjä. Korostamiseen käytettäisiin sen sijaan lämpimiä värejä, sillä nämä tuovat kontrastia muuten kylmään ulkoasuun ja ovat näin paremmin erotettavissa. Fonteissa tyyli olisi sarjakuvamainen, eli pyöreitä muotoja suosittaisiin. Osiona grafiista tyyliä oli hankala suunnitella ja dokumentoida, sillä kirjoittajalla ei taiteen suhteen ole merkittävästi kokemusta.

Graafisten assettien listalle tehtiin taulukko selkeyden vuoksi ja tavoitteena oli tehdä listasta sellainen, että eri aseteilla olisi jo valmiiksi määrittelyjä, joiden perusteella assetin voisi jopa tehdä. Lista lopulta koostui suurimmaksi osaksi käyttöliittymän 2D-grafiikkaa, sekä tietenkin 3D-asetteja. Nämä oli eri määrittelyjen vuoksi jaettu omiin taulukoihin. Määrittelyt 2D-grafiikan suhteen olivat selkeä nimi, kuvan koko, muunnelmät sekä muu tieto. Muunnelmilla tarkoitettiin, mikäli samasta grafiikasta tulisi useita versioita: painike-grafiikalle tulisi omat versionsa esimerkiksi perustilalle, sille kun hiiri on napin yllä ja sille kun nappia on painettu.

Määrittelyt 3D-grafiikan suhteen olivat taas selkeä nimi, tieto onko objekti staattinen vai dynaaminen, tarvittavat animaatiot, tieto onko objekti sellainen, että pelaaja voi kommunikoida sen kanssa ja muu tieto. Staattisuudella/dynaamisuudella viitataan siihen, että onko objekti vain paikallaan vai liikutellaanko sitä mahdollisesti ympäriinsä. Paikallaan olevaa objektia voi yksinkertaistaa, sillä sitä ei esimerkiksi nähtä tietystä kulmasta. Sen sijaan liikuteltava objekti on oltava kunnossa kaikista kulmista. Muussa tiedossa hyvin usein täsmennettiin tarve objektin rakenteen modulaarisuuden tarpeelle, eli sille, voiko objektin jotain erillistä osaa liikuttaa ympäriinsä. Tämä voisi olla esimerkiksi ovi-objektissa oleva oven kahva.

Äänien suhteen suunnittelu ja dokumentointi oli vielä vaikeampi mitä grafiikan suhteen, sillä tämä on kirjoittajalle tuntemattomin osa-alue. Tyylin suhteen oli kuitenkin järkevää pitää se grafiista ulkoasua mukailevana. Hämmäntävästä ajatusmaailmasta huolimatta paino olisi enemmän iloisen puolella ja musiikki olisi otteluiden aikana nopeatempoinen ja tyyliiltään piristävä. Sen sijaan valikojen aikana musiikki olisi rentoa ja hidastempoisempaa. Ääniefektit sen sijaan olisivat sarjakuvamaisia. Tällä tavoin tavoiteltu ajatusmaailma toteutuisi: visuaalien ja äänien suhteen peli on iloinen, mutta asetelma sen sijaan herättää kysymyksiä ja voidaan nähdä myös surullisena, mikä synnyttää kontrastia. Tämän kontrastin ja moninpelin kautta taas voi syntyä huumoria.

Mitä tuli ääni-asettien listaukseen, tässä seurattiin samaa menetelmää mitä grafiikassa. Assesteista tehtiin taulukkoon lista, jossa jokainen asetti sai selkeyden vuoksi lisämääritelmiä. Näitä oli esimerkiksi tieto siitä, toistetaanko ääni 2D vai 3D avaruudessa ja että toistetaanko ääni vain kerran, vai onko sen tarkoitus toistua silmukassa. Avaruusmääritelmä on hyödyllinen siksi, että tietää valmiiksi, onko äänen tarkoitus ottaa huomioon etäisyyksiä vai toistetaanko tämä tavallaan suoraan pelaajan korvaan. Koska tarkoituksena on käyttää FMOD-ohjelmistoa äänien ja musiikin tekoon, nämä tiedot antavat jo paljon tietoa siitä, mitä FMOD-ohjelmiston puolella kuuluu tehdä.

4.7 Kenttäsuunnittelu

Kuten aiemmassa osiossa todettiin, kenttäsuunnittelu jätettiin viimeiseksi suunnittelussa. Tämä siksi, että tämä vaatisi oman aivoriihensä. Tätä ei kuitenkaan voinut tehdä ensimmäisen aivoriihen aikaan. Vaikka idea ja mekaniikat olivat aivoriihen myötä selkeämpiä, varsinainen dokumentointi synnytti lopulliset yksityiskohdat sekä myös osoitti muutamia ongelmakohtia. Kenttäsuunnitelmat olisi joutunut muuttamaan joka tapauksessa.

Tavoitteena kenttäsuunnittelun aikana oli suunnitella vain yksi alue, jossa pelaajat pelaisivat ottelunsa. Tämä siksi, koska toimiakseen peli tarvitsee vain yhden tason. Lisää kenttiä voi suunnitella taas myöhemmin ja tämän työn kannalta tämä säästää myös aikaa. Idea tälle ainoalle kentälle tuli ensimmäisestä pelistä, eli lumiukkojen laakso. Ensimmäinen tavoite oli tehdä kenttään muotoja ja nähtävyyksiä, ensimmäisen pelin tasainen metsä oli hyvin tylsä. Ikävä kyllä ufon vuoksi suunnitelmia joutui rajaamaan esimerkiksi tunnelien ja luolien suhteen, joskin ajatuksena oli tehdä tilapäisiä piilopaikkoja, jotka ufo pystyisi tuhoamaan. Näiden piilopaikkojen virkaa toimittaisivat iglut, joita voi ajatella lumiukkojen asuntoina. Muotojen suhteen kenttä rajoittui lähinnä korkeuseroihin ja muodoissa pyrin tekemään niistä visuaalisesti tulkinnanvaraisia, kuten yksittäisistä pilvistä muuten selkeällä taivaalla. Kentälle tehty vuori näyttää esimerkiksi kanelipullalta ja tämän vasemmalle puolelle sijoitettu linnake näyttää taas tyhmältä naamalta. Näiden välillä oleva silta taas kahden iglun kera muodostaa kieltä näyttävän naaman ja jäinen järvi muistuttaa mahalaukkuja. Ikävä kyllä ensimmäisestä persoonasta nämä muodot saattavat mennä hukkaan. Lumiukkolaaksoon tuli myös eteläpuolelle kaivos, jota voi ajatella lumiukkojen työmaana. Vaikkei luolia olisi mahdollista tehdä, raiteita pystyy laittamaan seinämien yhteyteen ja näin ollen korkeuserot olisivat silti mahdollisia.

Asia mitä aluksi kenttäsuunnittelussa välteltiin, oli tapahtumat ja objektit, joiden kanssa voi kommunikoida. Tämä siksi, että näiden toteutuksen realistisuus moninpeliympäristössä olisi epävarmaa. Myöhemmin asiaa mietittiin uudelleen ja todettiin, että tapahtumien ja hallittavien objektien sisällyttäminen suunnitelmaan ei kuitenkaan ole huono asia, näiden toteutuksen mahdollisuuksia voi tutkia ja mikäli nämä eivät olekaan mahdollisia, nämä jättäisi pois. Tärkeintä on kuitenkin tehdä kentästä mahdollisimman hauska ja mielenkiintoinen. Tämän myötä syntyi ideoita esimerkiksi voimien ja työkalujen suhteen: näitä voisi käydä poimimassa kaivoksen edustalta tai pilkkireiästä järven jäällä. Järven jäällä voisi myös olla halkeamia, joihin astuminen aiheuttaa avantoon uppoamisen, mikä taas pysäyttää pelaajan hetkeksi.

Kenttäsuunnittelussa oli aiemman aivoriihen tapaan menetelmänä piirtää ajatukset paperille. Lopputuloksena aivoriihen päätteeksi olikin kartan oloinen piirros lumiukkolaaksosta, johon tärkeät asiat oli korostettu nuolin ja tekstein. Dokumentaatiossa kenttäsuunnittelun rakenteessa oli tavoitteena, että jokainen kenttä saisi oman osionsa. Tämä osio taas koostuisi kentän kartasta, sekä tekstistä joka selventää kartan sisältöä. Kenttäsuunnittelun jälkeen grafiikka-asettien listaa oli vielä päivitettävä.

4.8 Lopuksi

Kenttäsuunnittelun myötä pelisuunnitteludokumentti olisi nyt valmis. Rakenne suunniteltiin, pelin olemusta mietittiin aivoriihenä ja lopulta yksityiskohdat dokumentoitiin osio osiolta. Tämän jälkeen dokumentti luettiin vielä alusta alkaen useampaan kertaan ja tehtiin mahdollisia lisäyksiä yksityiskohtiin ja korjattiin vikoja. Lisäyksiä oli esimerkiksi ottelulogi, jossa siis näytetään tietoja mitä pelin aikana tapahtuu. Tämä oli sisällytetty käyttöliittymäsuunnitelmiin, mutta sitä ei mekaniikkana ollut selitetty ollenkaan. Myös ufon tuhokyvvyt loistivat poissaolollaan, mutta nämä lisättiin. Merkittäviä vikoja tarkistuksen aikana ei esiintynyt, mutta yksi oli esimerkiksi ajatusmaailman ristiriitaisuus. Tämä oli alun perin hauska ja iloinen. Graafisen ja äänimaailman tyylit pohjautuivat juuri tähän ajatusmaailmaan. Heräsi kuitenkin ajatus asetelman suhteen, se kun oli oikeastaan aika surullinen ja ehkä hämmentävä. Hauskuus ei kuitenkaan häviäisi mihinkään, tämä kun oli läsnä myös ensimmäisessä pelissä. Ajatusmaailman tekstiä oli kuitenkin muokattava vähän mukailemaan todellisuutta.

Tämän myötä pelin suunnittelu saatiin näin alustavasti päätökseen. Pelisuunnitteludokumentti tulee elämään kehityksen aikana sekä ideoita pelin suhteen saattaa tulla myöhemmin, joten on turha tässä vaiheessa lyödä suunnitelmien kaikki osa-alueet lopullisesti lukkoon. Prototyypin myötä on myös katsottava missä suunnitelmissa on paranneltavaa. Seuraavassa osiossa raportoidaan tämän pelisuunnitteludokumentin pohjalta tehdyn prototyypin teko.

5 JATKO-OSAN PROTOTYYPIN KEHITYS

Tässä osiossa on tarkoitus raportoida työn toisesta käytännön osuudesta, prototyypin teosta. Prototyyppi tehdään aiemmin tehdyn pelisuunnitteludokumentin pohjalta. Työn aikana pidettiin päiväkirjaa, johon päivittäin tehdyt asiat ja ongelmat kirjattiin muistiin. Raportti on tehty tämän päiväkirjan pohjalta. Lopputuloksena syntyi toimiva prototyyppi.

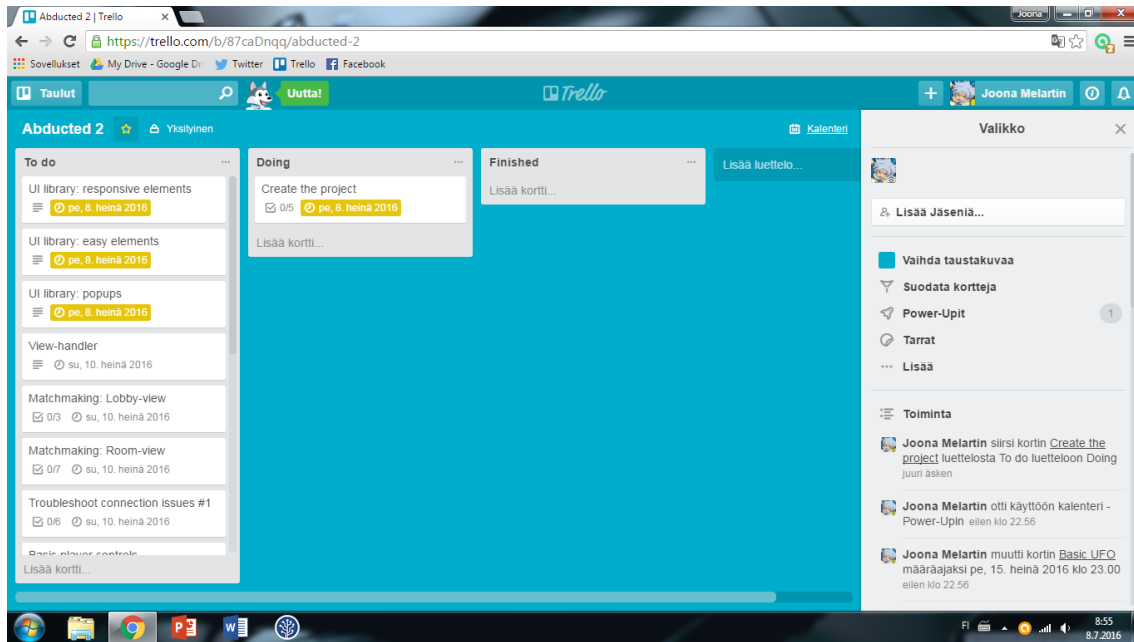
5.1 Rajaus ja valmistelut

Ennen kuin prototyyppiä lähdettiin tekemään, oli tärkeää ensin päättää mitä kaikkea prototyyppiin halutaan sisällyttää. Tämä siksi, että koko pelin teko suunnitelman pohjalta olisi aivan liian työläs tätä opinnäytettä ajatellen. Lähtökohtaisesti prototyypistä ei haluttu tehdä isoa, sillä alkuperäisen pelin idea muuttuu merkittävästi. Suunnittelun kannalta on siis järkevä tehdä pieni käytännön toteutus suunnitelmista ja katsoa toimiiko se käytännössä.

Prototyyppi rajattiin seuraaviin ominaisuuksiin:

- PC koontiversio
- Satunnais- ja yksityispelien teko
- Hallinta odotushuoneessa ajastimen kanssa
- Pelaajan kontrollit
- Ottelurakenne, tuloksien näyttäminen ja palautuminen
- Ufon tekoäly
- Työkalut: spring boots, snowboard, target gun, water gun
- Yhteysongelmista palautuminen
- Ei ääniä
- Vain välttämättömin grafiikka

Prototyyppi ei sisältäisi siis mitään näiden ominaisuuksien ulkopuolelta. Mainittakoon, että koontiversio poikkesi suunnitelmista siksi, että työhön ei löytyisi tekniikkaa tehdä selainversiota. Lisäksi prototyypin testailu työn aikana olisi helpompaa, sillä selainversio pitäisi erikseen upottaa nettisivulle.



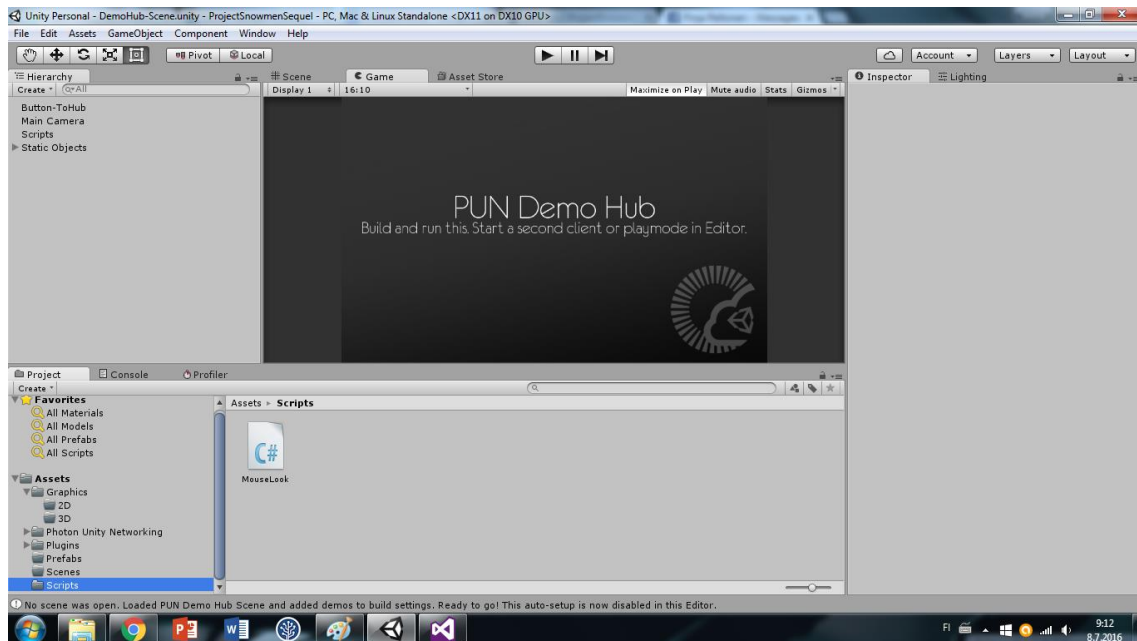
Kuvio 11 Prototyypin Trello-alusta

Kun prototyyppi oli rajattu, oli seuraavana vuorossa työn valmistelut. Selkeyden vuoksi prototyypille tehtiin Trello-sivustolle oma alusta, johon työhön liittyviä tehtäviä voisi hallinnoida ja aikatauluttaa. Kyseinen sivusto on suuressa käytössä erityisesti ketterässä ohjelmistokehityksessä ja on myös kirjoittajalle ollut apuna useiden projektien organisoinnissa. Aikataulu oli aluksi hyvin löysä, mutta kun työ eteni, tätä kiristettiin aina kun mahdollista. Hyvin usein arvio ominaisuuden valmistumisesta oli kauempana kuin mitä sitten todellisuudessa.

Aikataulun lisäksi prototyypin tehtävät tuli myös asetettua tärkeysjärjestykseen. Jos tehtäviä ajatellaan korkealla tasolla, järjestys oli seuraava:

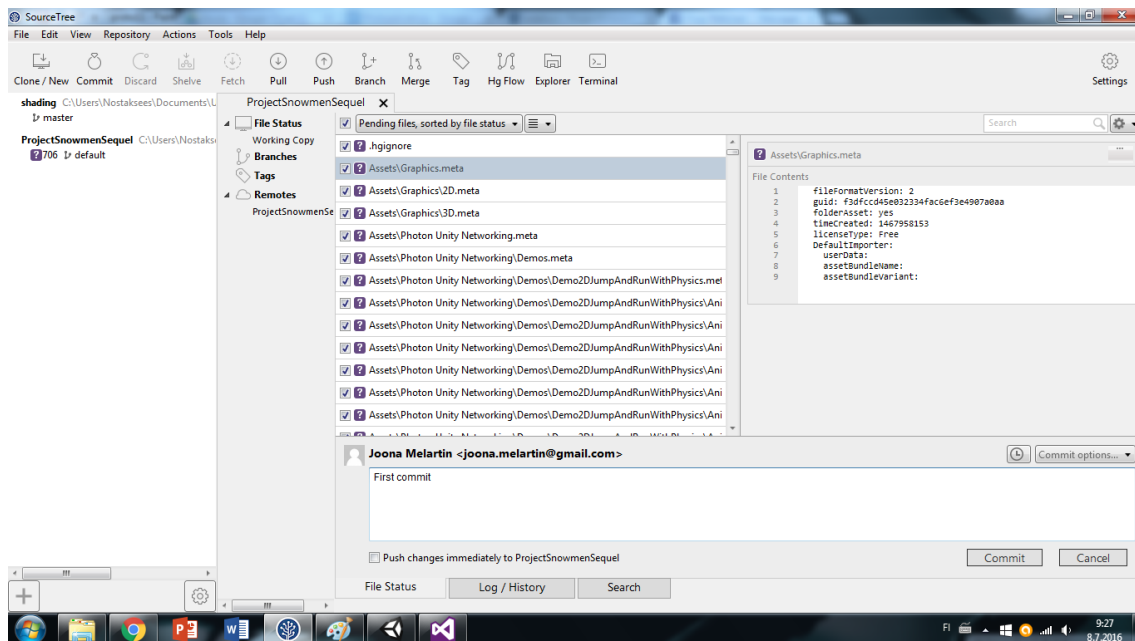
1. Projektin luonti, hierarkian organisointi ja lisäosat
2. Käyttöliittymän responsiivisuuskirjasto
3. Käyttöliittymän helppokäyttökirjasto
4. Popup-ilmoitukset
5. Otteluiden järjestäminen
6. Pelaajan kontrollit ja luonti, sivusta seuraaminen
7. Ufon tekoäly
8. Pelaajien ja ufon vaikutus ottelurakenteisiin
9. Yhteysongelmat ja niistä palautuminen
10. Työkalut

Periaatteena oli tehdä tärkein ensin. Valmistella ensin kaikki kuntoon ja sitten alkaa tehdä prototyyppin osia niin, että ydinasiat tehdään ensin. Järjestyksessä oli myös tärkeää, että ominaisuudessa tarvittavat muut ominaisuudet olisi tehty tätä ennen. Siksi esimerkiksi käyttöliittymän ohjelmointi on oikeastaan ensimmäisenä.



Kuvio 12 Photon Unity Network -lisäosa käyttövalmiina

Kun työ saatiin organisoitua, eli tiedettiin missä järjestyksessä ja millä aikataululla asioita aletaan tehdä, siirryttiin varsinaisen työn valmisteluihin. Tämä kattaisi siis ohjelmiston ja työkalujen valmistelun niin, että työn konkreettista osaa voisi alkaa tehdä. Tämä ei ollut iso työ, sillä ainoa oleellinen työkalu prototyyppiä varten on Unity 3D -ohjelmisto. Projekti ei vaatinut sen erityisempiä järjestelyjä kuin Photon Unity Network -lisäosan asentamisen. Tässäkin lisäosa vain tarvitsi ladata ja tähän piti kirjautua sisään, missä ei esiintynyt ongelmia. Tämän lisäksi prototyyppille tehtiin oma versiohallinta jatkokehitystä sekä varmuuskopiointia ajatellen Bitbucket-sivustolle. Versiohallintaa käytettiin taas SourceTree-ohjelmistolla. Nämäkin valmistelut olivat hyvin suoraviivaisia, ongelmia ei esiintynyt.



Kuvio 13 Versiohallinnan ensimmäinen commit

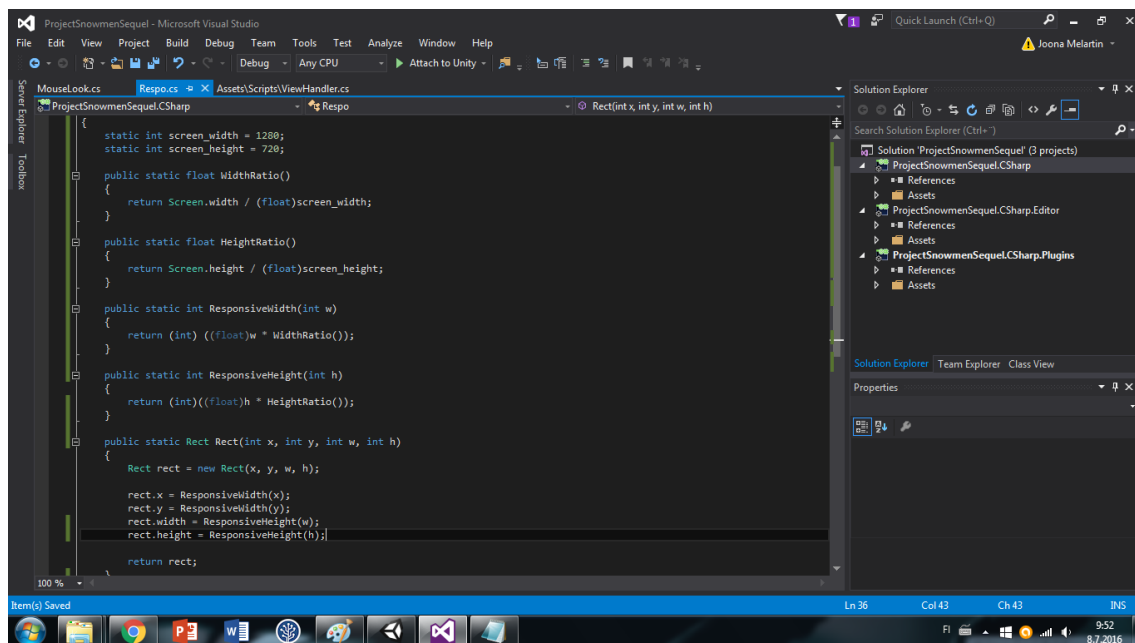
Rajauksen ja valmistelujen myötä prototyyppi oli nyt valmis tuotantoa varten. Tätä käydään läpi seuraavassa osassa.

5.2 Tuotanto

Ensimmäisenä prototyypille oli luotava käyttöliittymä. Tämä siksi, että melkein kaikki pelissä tapahtuu käyttöliittymän kautta ja ilman valmista järjestelmää sille on vaikea tehdä mitään peliin tai edes prototyyppiin. Teon kannalta käyttöliittymältä haluttiin yksinkertaisuutta ja vaikka Unity 3D -ohjelmisto tarjoaa oman hyvin helposti käytettävän UI-järjestelmän, päätettiin tehdä oma järjestelmä Unityn vanhaa GUI-järjestelmää hyödyntäen. Tämä GUI-järjestelmä tarkoittaa siis koodikirjastoa, jolla ruudulle voidaan piirtää painikkeita ja tekstiä. Nykyään tämä ei ole niin yleisessä käytössä uuden paljon graafisemman vastineen vuoksi. Koodikirjastolla taas tarkoitetaan kokoelmaa kooditiedostoja, joiden sisältämää koodi on suunniteltu käytettäväksi muussa koodissa.

Pääasiassa tämä ratkaisu omasta käyttöliittymäjärjestelmästä pohjautui kirjoittajan omaan mieltymykseen, mutta koska aiempaa kokemusta koodipohjaisesta käyttöliittymästä löytyy, tämä olisi oikeastaan nopeampi tapa. Pelin mekaniikkoja ja käyttöliittymää saisi helposti upotettua keskenään ja kunnon hallintajärjestelmän kanssa se olisi myös helppo hallita.

Jotta tämän järjestelmän saisi tehtyä, oli ensimmäiseksi korjattava eräs vika Unityn vanhassa käyttöliittymäjärjestelmässä: responsiivisuuden puute. Responsiivisuudella tarkoitetaan sitä, että käyttöliittymä osaa muuttua ruudun koon mukaan. Esimerkiksi verkkosivut on nykyään hyvä tehdä responsiivisiksi, sillä ihmiset selaavat verkkoa hyvin erikokoisilta ruuduilta. Peleissä tämä helpottaa merkittävästi tuotantoa ja varmistaa myös, että eri näytöillä ei syntyisi merkittäviä eroja. Unityn vanha käyttöliittymäjärjestelmä onkin sellaisenaan täysin staattinen; mikäli responsiivisuutta haluaisi, on tämä laskenta tehtävä itse. Tavoitteena onkin tehdä koodi, joka tekisi tämän ohjelmoijan puolesta.

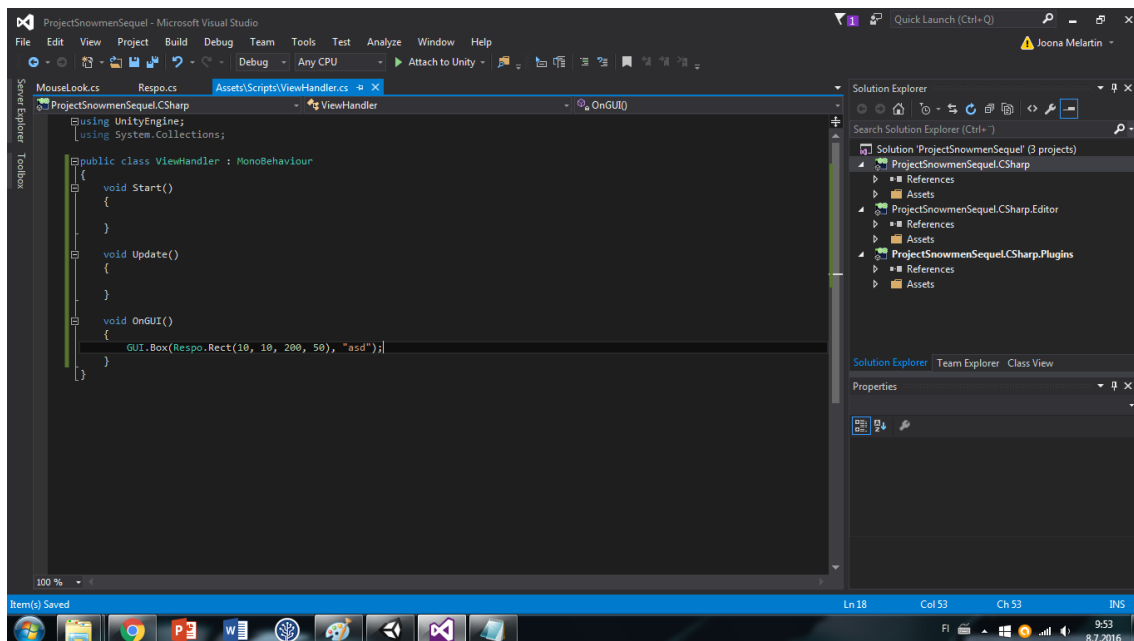


Kuvio 14 Responsiivisuus-kirjasto

Kuviosta 14 voi nähdä pätkän koodia, jolla responsiivisuus on toteutettu. Huomioitavaa on, että jokainen funktio koodissa omaa static-etuliitteen. Tällä tarkoitetaan sitä, että kyseinen muuttuja/funktio varaa tietyn kohdan tietokoneen muistista, mikä mahdollistaa sen, että tähän muistiin voi viitata mistä tahansa itse sovelluksessa. Kääntöpuolena on taas se, että tätä muistia ei voi muokata yhtä vapaasti.

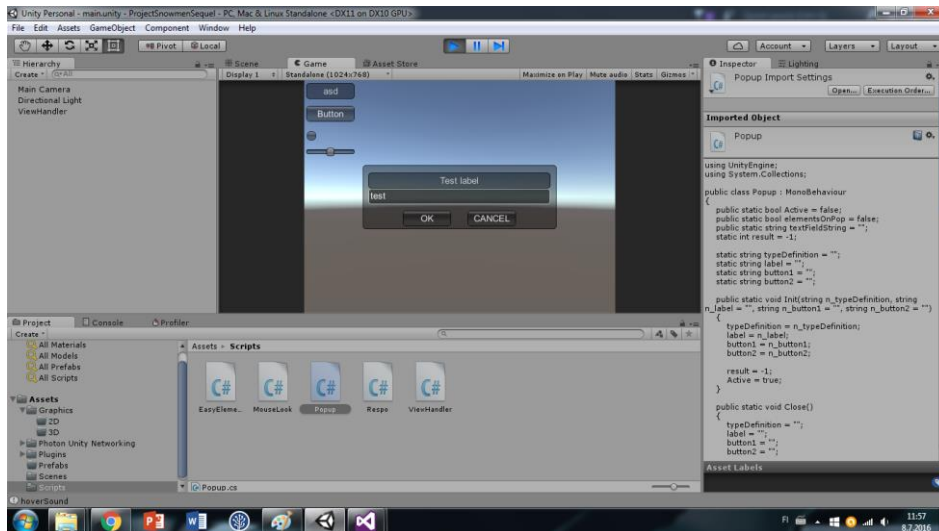
Static-etuliitteen virka tässä yhteydessä on se, että tätä koodia tarvitseva toinen koodi ei tarvitse erillistä muuttuja-viittausta tähän koodiin. Muuttuja-viittaus olisi ainoa keino päästä käsiksi tähän koodiin, mikäli funktiot ja muuttujat ei olisi tehty static-etuliitteen kanssa. Tämä helpottaa merkittävästi koodin kirjoittamista ja myös nopeuttaa hiukan laskentaa.

Lopulta tämä koodikirjastoksikin nimitettävä koodi alkoi toimia seuraavalla periaatteella: käyttäjä määrittää ensin kohderesoluution, missä peliä pelattaisiin. Tässä tapauksessa tähdättiin 720p HD-resoluutioon, eli 1280x720. Kun koodia sitten lopulta käytetään, annetaan koodikirjastolle funktion kautta halutut arvot. Kirjasto sitten laskee kohderesoluution mukaan uudet arvot ja antaa ne takaisin pyyntökohtaan. Myöhemmin kirjastoa laajennettiin tasaus-ominaisuuksilla. Koodin kautta pystyi antamaan esimerkiksi tekstimuodossa yhdelle akselille "center", mikä komentaisi koodia keskittämään tämän valitun akselin.



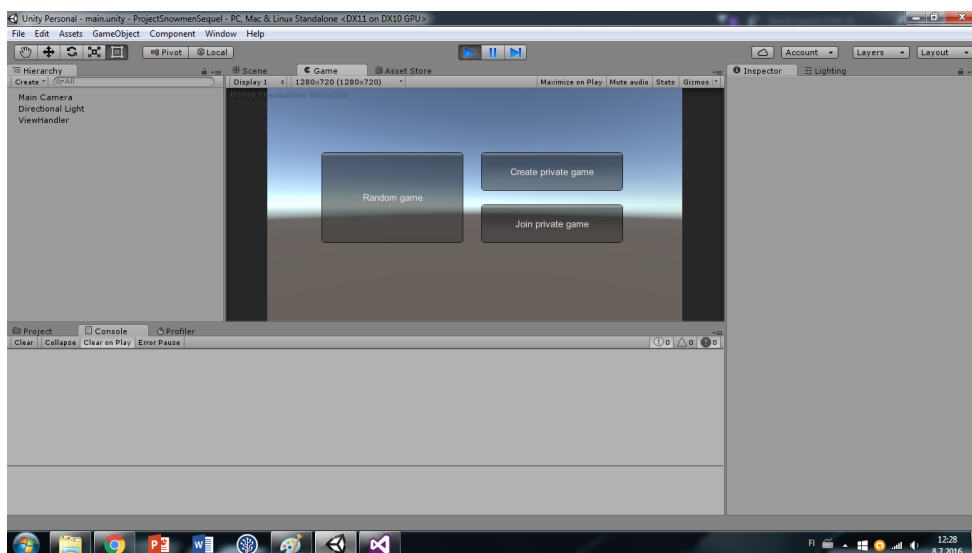
Kuvio 15 Responsiivisuus-kirjaston käyttöesimerkki

Responsiivisuus-kirjasto saatiin tehtyä ilman ongelmia, joskin tietotyyppien kanssa piti olla tarkkana. Tämän jälkeen oli vuorossa helppokäyttökirjasto Unityn GUI-funktioille. Nämä toimisivat sellaisenaan ihan hyvin, mutta mikäli ääntä ja tyyliä haluaa hallita järkevästi, helpottaa merkittävästi, että toinen koodi käsittelee tämän yhdessä paikassa. Lähestymistapa tähän oli oikeastaan sama kuin responsiivisuus-kirjastolla; staattisia funktioita, jotka palauttaisivat halutut arvot. Kirjasto rajattiin silti vain painikkeisiin, tarkistuslaatikkoihin ja liukusäätimiin. Tämän lisäksi tehtiin myös mahdollisuus ponnahdusikkunoille, joka aluksi vaikutti vaikeimmalta tehdä, mutta olikin ainoastaan työläin. Ponnahdusikkunat kuitenkin pitää näyttää aina päällimmäisenä, eikä näiden läpi pitäisi pystyä painelemaan mitään. Lopulta tämäkin valmistui ja toimi juuri kuten piti, tehden käyttöliittymäjärjestelmästä pitkälti valmiin käyttöä.



Kuvio 16 Aiemmin määritellyt koodikirjastot testikäytössä

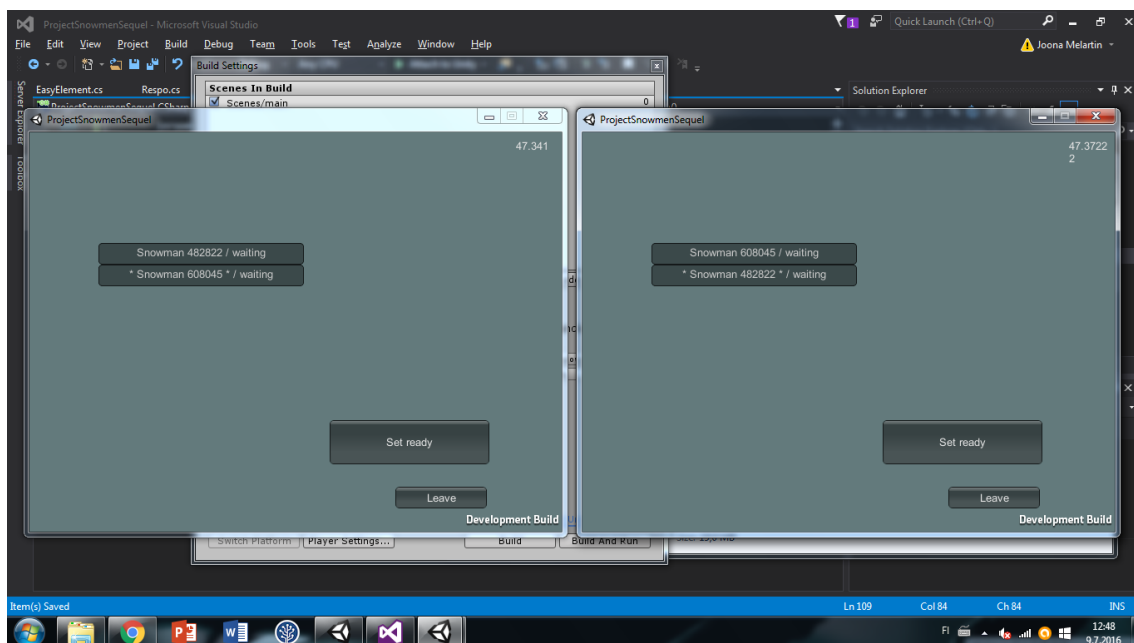
Seuraavana työjärjestyksessä oli alkaa tehdä itse pelin mekaniikkoja. Ennen pelaajan kontroleita taikka mitään mikä liittyisi itse pelin idean toteutukseen, oli ensin tehtävä moninpelimekaniikat. Tämä siksi, koska aikataulutuksen aikana ennakoitiin moninpelimekaniikkojen vievän aikaa. Oli siis ensimmäiseksi selvitettävä miten työläitä nämä oikeasti ovat. Puhumattakaan siitä, että Photon Unity Network -lisäosalla ei mitään yksinpeliä voisi suoraan kääntää monipeliksi. Pelaajan luonti toimii esimerkiksi hyvin eri tavalla. Tässä mielessä moninpelimekaniikat tulisivat tärkeysjärjestyksessä siis joka tapauksessa ensimmäisenä.



Kuvio 17 Ensimmäinen näkymä

Tavoitteena oli aluksi tehdä vain mekaniikat peliin liittymistä varten. Pelaajat pystyvät valitsemaan minkälaiseen peliin haluavat liittyä ja seuraavassa näkymässä pelaajat voisivat nähdä toisensa.

Vaihtoehtoina pelien välillä on satunnainen ottelu tai yksityinen ottelu. Yksityiset ottelut eivät tulisi satunnaisia otteluita hakiessa ja nämä voisi suojata salasanalla, jonka voi antaa kavereilleen. Tässä ei ollut oikeastaan muuta ongelmaa kuin se, että aluksi satunnaisista peleistä pystyi hakemaan myös yksityisiä otteluita. Hyvin moni vastaava ongelma oikeastaan johtuikin siitä, että luoduille ”huoneille” välitettiin arvoja väärin tai hakemisen aikana välitetyt arvot annettiin hakuun väärin. Huoneella tarkoitetaan tässä yhteydessä tilaa, johon pelaajat ovat liittyneet ja jossa nämä odottavat varsinaisen pelin aloitusta.



Kuvio 18 Huone-näkymä

Seuraavaksi oli tarkoitus tehdä lisää ominaisuuksia huone-näkymään, mutta tätä hidasti juuri aiemmin kerrotut ongelmat niin huoneiden kuin pelaajien tietojen parissa. Kun näistä päästiin yli, keskityttiin lähinnä perusasioihin näkymässä: tietojen näyttäminen ja huoneesta poistuminen. Lisäksi tehtiin myös alustavat peli- ja tulospäätökset, joissa ei kuitenkaan tapahdu mitään. Tämän tarkoituksena oli lähinnä varmistaa navigoinnin ja tiedonsiirron toimivuus. Ensimmäinen iso ongelma kuitenkin ilmeni, kun huone-näkymässä käynnistettiin ajastin. Tämän loputtua varsinainen peli siis alkaisi. Ajastimen arvo ei siirtynyt muille ja ehdot joiden perusteella tämä käynnistetään eivät toimineet. Lisäksi näin PC koontiversiossa ajastimen kykeni sekoittamaan täysin liikuttamalla peliikkunaa ympäriinsä.

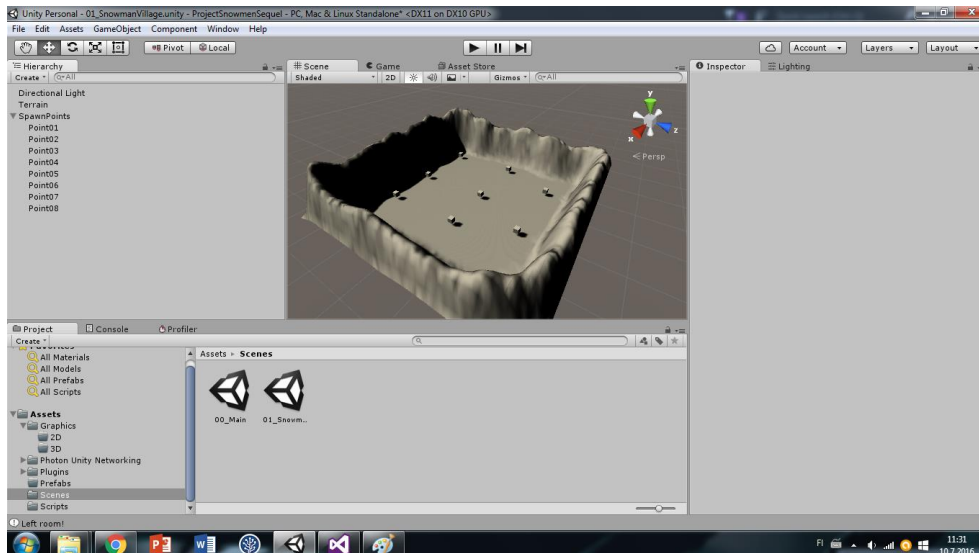
Jälkimmäiseen ongelmaan ei koskaan mitään korjausta löydetty, mutta toisaalta tällä ei suurta merkitystä olisi, lopullinen peli kun tulisi selaimelle. Muut ongelmat sen sijaan korjautuivat huolellisella

RPC-komentojen järjestämisellä. Nämä ovat siis käskyjä, jotka lähetetään verkon kautta muille ja joille löytyy vastaava funktio koodissa. Lopulta ajastin alkoi toimia suunnitellusti. Myös navigointi peli- ja tulospäivityksiin toimi oikein.

Tässä vaiheessa perusasiat monipelissä toimivat: pelaaja voi liittyä huoneeseen, poistua missä tahansa välissä pois ja ajastimet huolehtivat, että siirtymät peliin, tuloksiin ja sieltä pois tapahtuvat tarvittaessa pelaajista riippumatta. Seuraavana oli vuorossa varsinaisen pelin teko. Jotta pelaajat eivät putoilisi heti alussa tyhjyyteen, oli ensiksi tehtävä kartan luonti.

Suunnitelmana oli tehdä vain yksi kartta, mutta myös valmius sille, että useampi olisi mahdollinen. Tätä yhtä karttaa ei lopulta tehty sen pohjalta mitä pelisuunnitelmassa oli, sillä tämä olisi prototyypin kannalta turhan työlästä. Tässä kohtaa myös tajuttiin, että huoneissa on säilytettävä myös tieto siitä, mikä kartta olisi kyseessä. Kartta kun valitaan satunnaisesti, olisi mahdollista, että eri pelaajille tulee eri kartta. Tätä ei tietenkään haluttaisi. Aluksi ongelmaksi muodostui se, miten kartta oikeastaan liitettäisiin peliin. Tälle kuitenkin löytyi yksinkertainen ratkaisu, nimittäin Unity mahdollistaa scene-tiedostojen yhdistämisen pelin aikana. Scenellä tarkoitetaan siis tilaa, jossa pelin toiminnallisuus ja assetit sijaitsevat. Tätä voisi suoraan verrata videopelin tasoon.

Tämä scene-tiedostojen yhdistäminen mahdollistaa sen, että pelaajan ei itsensä tarvitse siirtyä scenejen välillä (mikä mutkittasi tiedonsiirtoa, Unity kun ei automaattisesti siirrä tietoja scenejen välillä) ja lisäksi eri karttojen organisointi olisi helpompaa. Jos asiaa ajattelee versiohallinnan kannalta, on kannattavampaa pitää kukin kartta omassa scene-tiedostossaan. Unityn scene-tiedostot eivät perinteisessä Git- tai Mercurial-versiohallinnassa yhdisty ollenkaan, mikä tarkoittaa, että yhtä scene-tiedostoa voi olla muokkaamassa vain yksi kerrallaan. Tällä järjestelyllä voidaan kiertää tämä ongelma ja jokaista karttaa voisi nyt muokata erikseen samanaikaisesti. Toinen vaihtoehto olisi ollut tehdä kaikki kentät yhteen scene-tiedostoon ja sitten asettaa vain yksi niistä päälle, mutta versiohallinnan kannalta tämä ei olisi toiminut yhtä hyvin.



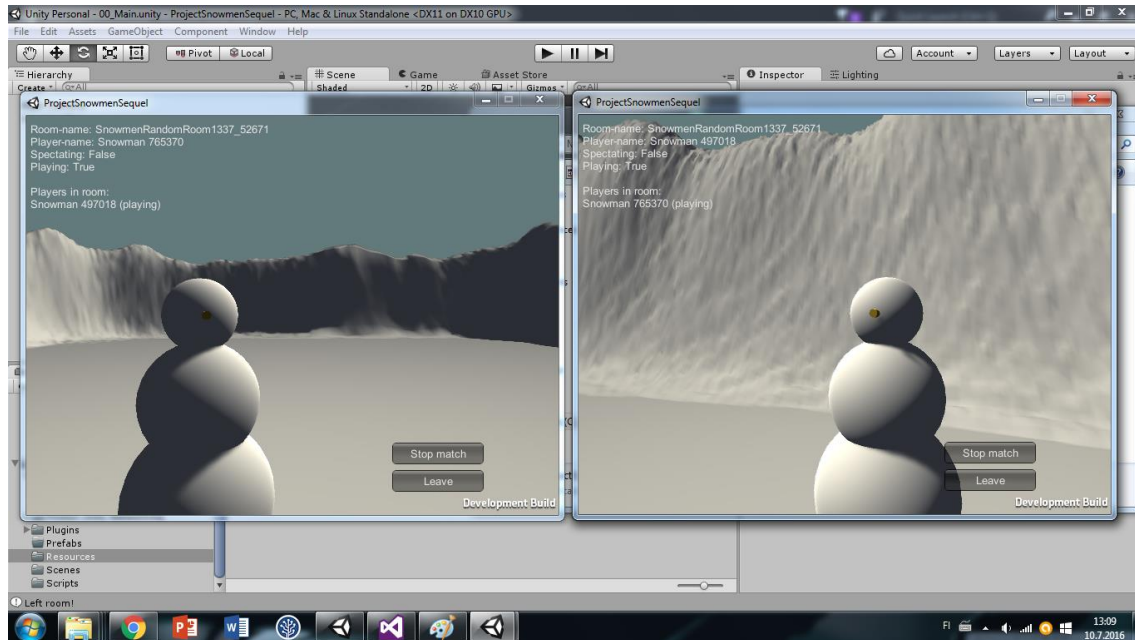
Kuvio 19 Prototyypin kartta

Lopulta satunnaisen kartan lataaminen alkoi toimia ja tämä tieto myös saatiin säilymään samana kaikille. Tästä seurasi kuitenkin ongelmia esimerkiksi sivustakatsojien suhteen: kun nämä liittyivät peliin, nämä jäivät jumiin kartan lataukseen. Asia selvisi kuitenkin tutkimalla tarkkaan koodia, nimittäin sivustakatsojan liittymisestä puuttui ominaisuuksia, joita normaalilla pelaajalla oli. Tämäkin saatiin siis toimimaan. Sivustakatsojalla tarkoitetaan tässä yhteydessä siis pelaajaa, joka on mukana pelissä mutta vain katsojan roolissa.

Tämän jälkeen ongelmia alkoi esiintyä tilanvaihdon kanssa. Jos pelaaja siis liittyy peliin sivustakatsojana tuloksien aikana, tuloksien katselun jälkeen tämä ei siirry huone-näkymään vaan jää oikeastaan jumiin tuloksiin. Ajastimen nollautuessa tämän jälkeen pelaaja siirretään huone-näkymään. Muut pelaajat kuitenkin tällä hetkellä ovat siirtyneet jo peliin. Vika oli lopulta oikeastaan ajastimessa. Ajastin nimittäin teki asioita liian varhain, jolloin pelaajamäärät ja ylipäättänsä elementit käyttöliittymässä eivät vielä olleet ajan tasalla. Ongelma ratkesi, kun ajastinta pakotettiin odottamaan oikeaa hetkeä. Tämä tapahtui tarkistamalla suoraan kaikkien huoneessa olevien pelaajien tila.

Ongelmien selvityksen jälkeen oli taas aika siirtyä eteenpäin. Nyt kun kartta saatiin ilmestymään peliin, oli aika tehdä tänne aloituspisteet sekä tietenkin itse pelaajat. Tämä tapahtui aluksi perusasioiden kautta: ensiksi pelaajat ilmestyivät peliin pelkkinä kameroina ja lopulta näille lisättiin liikkumismahdollisuus sekä ruumis, joka näkyy muille. Samalla tehtiin myös sivustakatsojien mekaniikka.

kat sekä tavallisen pelaajan ja sivustakatsojan prefabin luonnin erottelu. Prefabilla tarkoitetaan asetta, joka sisältää pelin toiminnallisuutta ja muita asetteja. Pelaajan prefab koostuisi esimerkiksi kamerasta, ruumiin 3D-mallista ja koodista, joka hallinnoi pelaajan liikettä.

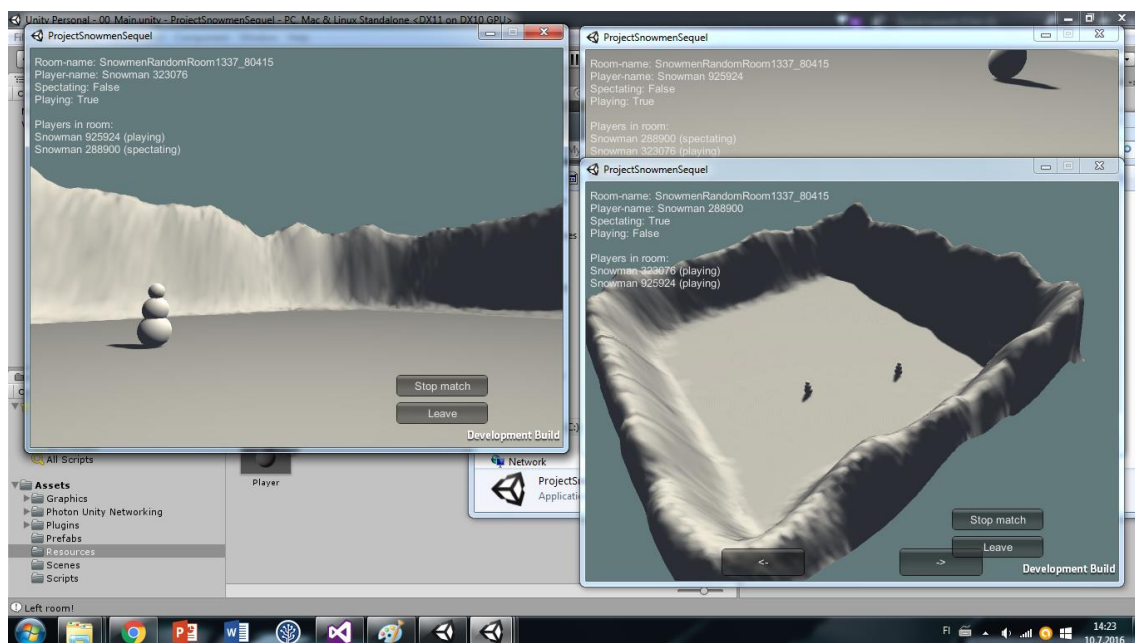


Kuvio 20 Pelaajat näkevät toisensa

Loppujen lopuksi tässä ei esiintynyt oikeastaan muita ongelmia kuin se, miten lumiukkojen ylimmän pallon piilottaisi. Tämä kun on täsmälleen samassa kohdassa kuin kamera, pelaaja näkisi esimerkiksi koristeita ja lumiukon nenän sisäpuolelta. Tähänkin löytyi oikeastaan yksinkertainen ratkaisu, mikä liittyy paljon ihan pelaajan luontiin verkossa. Järkevintähän pelaajan luonti on tehdä niin, että pelaajan mekaniikkoja hallinnoiva koodi tarkistaa, onko tällä omalla puolella verkkoa oleva pelaaja oma pelaaja. Tämän perusteella voidaan sitten erotella, miten pelaaja käyttäytyy itse pelaajan käsissä ja muiden pelaajien silmissä. Varsinaisen pelaajan puolella tarkkaillaan näppäimistön syötettä ja lähetetään tulokset muille. Muiden puolella tulokset taas vastaanotetaan ja saaduilla tiedoilla päivitetään pelaajan tilaa. Lisäksi kun pelaajan mekaniikat on asetettu oletuksena suljetuiksi prefabia luodessa, voidaan esimerkiksi mahdollistaa se, että vain yksi kamera on aktiivinen. Samalla periaatteella pelaajan ruumiin, eli lumiukon ominaisuuksia voitiin hallita.

Tämän jälkeen selvisi, että sivustakatsojien mekaniikoissa on oikeastaan vikaa. Sivustakatsojien on siis mahdollista liikkua pelkkänä kamerana pelissä ja näin seurata peliä, mutta vain niiden pisteiden välillä, mitkä on ennalta määritelty. Näitä pisteitä ovat kaikki aktiiviset pelaajat sekä karttaan upotetut valmiit pisteet. Ongelmaksi koostui se, että sivustakatsojan mekaniikat eivät onnistuneet

löytämään näitä pisteitä oikein ja näiden puhdistelu pelin edetessä ei toiminut. Puhdistusta tarvitaan, sillä eliminoidut pelaajat muuttuvat sivustakatsojiksi. Sivustakatsojien ei taas ole mitään järkeä seurata muita sivustakatsojia. Lisäksi sivustakatsojilla esiintyi sama ongelma pelaajan ruumiin yläosan kanssa kuin aiemmin. Kamerapisteisiin liittyvä ongelma johtui lopulta vääränlaisesta hausta ja puhdistelu alkoi toimia, kun käytettiin oikean haun lisäksi oikeanlaista tietotyyppiä. Ruumiin yläosaan liittyvä ongelma taas ratkesi sillä, että pelaajan mekaniikkoihin lisättiin julkiseksi muuttujaksi viittaus ylimpään palloon. Sivustakatsojan mekaniikat taas pystyivät käyttämään tätä muuttujaa piilottaakseen seuratun pelaajan ylimmän pallon, sekä tietenkin näyttämään tämän uudelleen poistuttaessa.



Kuvio 21 Kolme pelaajaa, joista yksi on sivustakatsoja

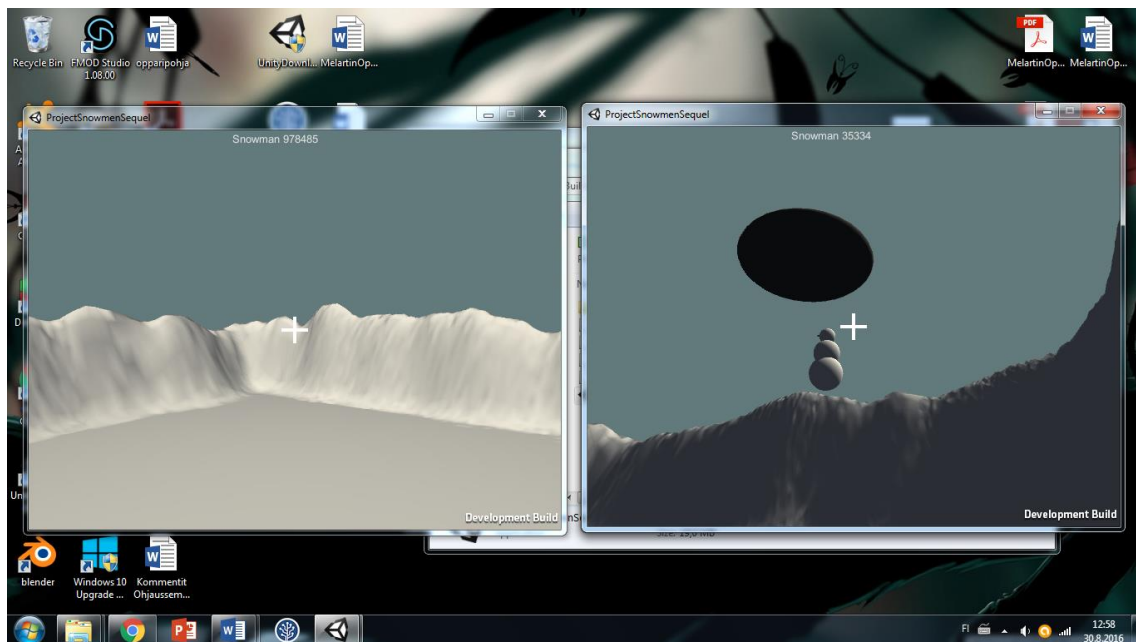
Tässä vaiheessa kukin pelaaja pystyi pääsemään pelin kartalle, liikkumaan ja katselemaan ympärilleen. Lisäksi kesken kaiken peliin liittyessä tämä pystyi seuraamaan peliä liikkumalla pelissä olevien kameroiden välillä. Seuraavana oli vuorossa tehdä ufon mekaniikat, mikä oli siis ainoa puuttuva asia, jotta varsinainen peli jo toimisi.

Ensimmäinen ongelma ufossa oli jo sen luonti. Sen luonti samalla tavalla kuin pelaajan luonti ei toiminut, sillä ufon luoneen pelaajan poistuessaa myös ufo häviäisi. Ufon luonti jokaiselle pelaajalle paikallisesti ei taas toiminut, sillä ufo käyttäytyi aivan eri tavalla joka pelaajalle. Lopulta Photon Unity Network -lisäosan koodikirjastoista löytyi funktio, jolla luonnin pystyi suorittamaan verkon yli

ja niin, että tämän olemassaolo ei olisi pelaajariippuvainen: PhotonNetwork.InstantiateSceneObject. Tämän myötä ufon varsinainen luonti alkoi toimia.

Seuraava ongelma oli se, miten pelaaja saisi tiedon kaikista pelaajista. Näistä luonnollisesti kannattaisi pitää yllä listaa, mutta luontiviiveiden takia ufo ei saanut kuin yhden pelaajan listaan. Tämä ratkesi lopulta niin, että koska Photon pitää yllä listaa pelaajista ja tähän listaan on mahdollista kytkeä kartalla oleva objekti, ufo muodostaisi oman pelaajalistansa täältä. Ufo ei tekisi mitään, ennen kuin lista on saatu muodostettua.

Tässä välissä ilmeni eräs toinen ongelma, mikä liittyi taas sivustakatsojiin: kun nämä liittyvät peliin, nämä näkivät kartan ja pelaajat aivan eri tavalla. Tämä johtui siitä, että kartan luonnissa esiintyi viivettä. Vaikka Photon saa verkon yli luodut objektit näkyville melkein heti, scene-tiedoston tuonnissa kestää aikaa. Pelaajilla on hyppytoimintoa varten fysiikkamekaniikat (Rigidbody), joten ilman karttaa nämä alkoivat vain pudota alas sivustakatsojan näkökulmasta, vaikka näiden pelaajat lähettivätkin tietoa näiden oikeista sijainneista. Ratkaisu oli sulkea fysiikkamekaniikat kokonaan muiden kuin oman pelaajan kohdalta. Sijaintitietojen lähetyksen takia fysiikoita ei muiden pelaajien kohdalla nimittäin tarvittaisi.

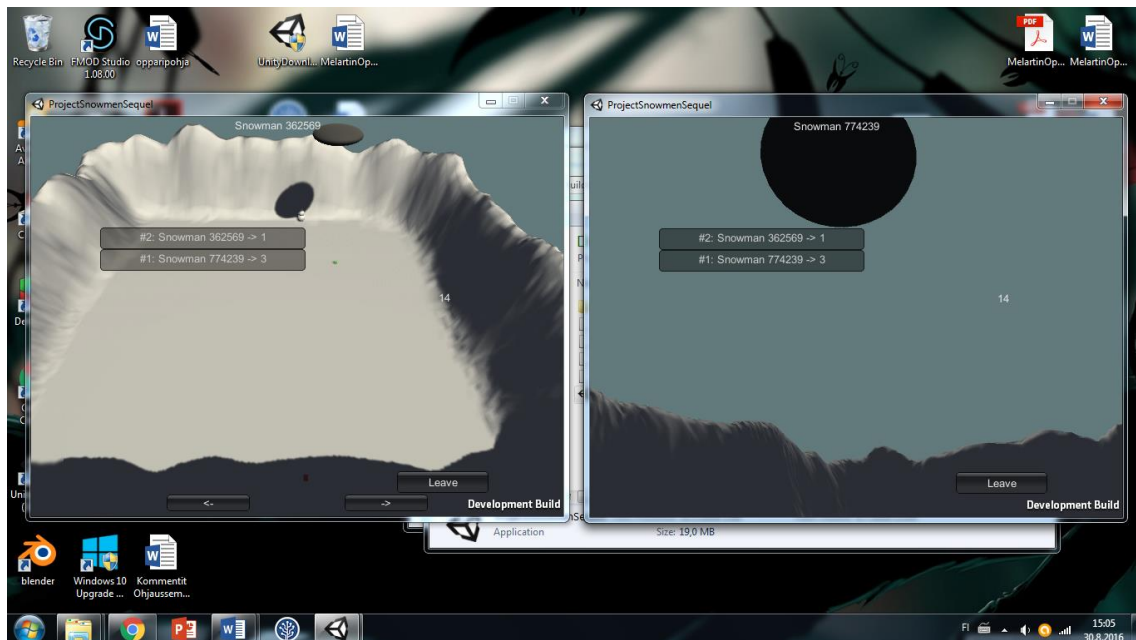


Kuvio 22 Ufo toiminnassa

Näiden ongelmien ratkaisun jälkeen kaikki toimi taas odotetusti. Seuraavaksi oli aika työstää ufon tekoälyä, tällä kun oli nyt valmiudet siihen. Ensiksi tekoälyyn tehtiin pelaajien seuraaminen sekä

kohteen vaihtaminen. Tämä tapahtui aiemmin mainitun pelaajalistan avulla: tietyin aikavälein ufo valitsisi satunnaisen kohteen ja liikkuisi tätä kohti. Kun tämä toimi, tehtiin ufon korkeushallinta. Ufo siis pysyisi aina samalla etäisyydellä kartan maastosta, vaikka täällä olisi kuoppia tai kukkuloita. Tämä siksi, että pelaajien eliminointiin menisi aina sama aika. Tämän jälkeen, kun huomattiin, että näissä ominaisuuksissa ei ollut suurempia ongelmia, tehtiin toiminnallisuus pelaajan kanssa. Ufo siis estää pelaajan liikkumisen ja alkaa vetää tätä itseään kohti. Tämäkin alkoi toimia, mutta ei täysin ongelmitta: ufo ei osannut lähteä uudestaan liikkeelle ja kun pelaaja vaihtuu sivustakatsojaksi, tämä ei saanut hallintaa hiirestään. Ensimmäinen ongelma ratkesi tietotyypimuunnoksella sekä yhden sekaannuksen selvittämällä koodissa. Hiiren hallinta palautui, kun tämän ominaisuuden lisäsi.

Seuraavaksi oli vuorossa tuloksien aktivointi. Tähän mennessä pelien päättämiseksi oli testaamisen takia oma nappinsa, joten tämä nappi oli vain korvattava oikeanlaisella tarkistuksella. Tunnistus tapahtui niin, että pelaajien lähtiessä tai muuttuessa sivustakatsojiksi, tarkistettiin, onko aktiivisia pelaajia vain yksi. Mikäli näin olisi, tulokset aktivoitaisiin ja kaikkien pelaajien kontrollit ja näkymät muutettaisiin sen mukaisesti. Tämä alkoi toimia ja oli aika tehdä varsinainen tuloslista. Tähän mennessä, vaikka näkymä on ollut olemassa, tämä ei ole sisältänyt kuin ajastimen huone-näkymään siirtymistä varten.



Kuvio 23 Tuloksien katselu

Tuloslista toteutui seuraavasti: kun pelaaja muunnetaan ufon toimesta sivustakatsojaksi, lisätään kaikille yhteiseen listaan RPC-komentojen kautta tieto siitä, kuka nyt putosi pelistä. Kun tulokset aktivoidaan, lisätään vielä jäljellä oleva pelaaja listaan. Lopputuloksena on lista pelaajista eliminointijärjestyksessä. Tämä lista käännettiin sitten toisinpäin ja esitettiin tulostuloksessa. Pelin ns. valuutta lisättiin tässä välissä ja tuloslistaan yhdistettiin myös tieto siitä, kuinka paljon kukin pelaaja saa tätä valuuttaa. Valuuttojen jako tapahtuisi taas paikallisesti yksinkertaisen tarkistuksen kautta.

Seuraavana oli vuorossa viimeinen suunniteltu prototyypin ominaisuus: työkalut. Tämä oli kenties monimutkaisin asia tehdä koko prototyyppiin. Työkalut nimittäin käsittivät näiden pudottamisen kartalle, pelaajan törmäyksen tunnistamisen, niiden esittämisen pelaajan ruumiin yhteydessä, niiden käytön ja vaikutuksien teon sekä vaikutukset käyttöliittymään. Erityisen vaikeaa oli pelaajaan liittyvät asiat, sillä tiedon synkronointi moninpeleissä ei ole ihan yksinkertaista, etenkin jos haluaa toimia isä-lapsi hierarkian kanssa. Lopulta usean mutkan kautta saatiin aikaan lopputulos, jossa pelaajan törmäys sekä työkalun siirtäminen pelaajan ruumiiseen alkoi toimia. Yksi osa tätä ratkaisua oli siirtää poimitun työkalun objektin omistajuus pelaajalle, jolloin objektin hallinta oli vapaampaa.

Kun työkalut alkoivat toimia, oli prototyyppi aika lailla valmis. Toimivuuden kannalta tähän toteutettiin kuitenkin pienimuotoinen testaus, mikä tapahtui vain pelaamalla prototyyppiä. Tämän aikana ilmeni useampikin ongelma, erityisesti tuloksien näyttämisen tunnistamisessa. Tässä oli esimerkiksi se ongelma, että jos pelaaja poistui sulkemalla pelin, ei tunnistusta tapahtunut. Vaikutti muutenkin, että tunnistus ei toiminut joskus normaalillakaan tavalla. Tämä ratkesi sillä, että tunnistuksen muunsi jatkuvaksi, aiemmin kun se tapahtui liian harvoin. Muita ongelmia esiintyi esimerkiksi synkronoinnin tarkkuudessa ja pelaajien oudoissa äärimajoissa vastapuolelta katsellen. Loppujen lopuksi nämä ongelmat saatiin korjattua, joskin perusteellisemmän testauksen tuloksena varmasti olisi löytynyt lisää ongelmia. Prototyypin kannalta tämä oli kuitenkin tarpeetonta ja tämän myötä prototyypin kehitys saatiin päätökseen.

6 TYÖN LOPPUTULOS

Tässä osiossa käydään läpi tämän opinnäytteen työn tuloksia. Tarkoituksena on tarkastella aiemmin tehtyä pelisuunnitelmaa sekä prototyyppiä jatkokehityksen kannalta. Lopuksi tarkastellaan ja pohditaan työtä ja sen prosessia yleisesti.

Kun tarkastellaan ensiksi tehtyä pelisuunnitteludokumenttia, ei voida oikeastaan todeta siinä olevan erityisiä puutteita. Kehitys onkin tässä tilassa vielä niin varhaisessa vaiheessa, että puutteita ei voikaan vielä todeta. Näillä näkymin pelisuunnitteludokumentti kuitenkin kattaisi sisällöllisesti pääosan kysymyksistä, joita kehityksen aikana voisi ilmetä. On myös todettava, että pelisuunnitteludokumentilla on turha hakea täydellisyyttä, se kun voi tulla vielä muuttumaan paljonkin varsinaisen kehityksen aikana. Pohjatyötä voisi katsoa olevan erittäin hyvin. Ei ole myöskään nähtävissä näin jälkeenpäin, että pelisuunnitteludokumentissa olisi pitänyt mitään tehdä toisin.

Varsinaista pelin ideaa on vaikea arvioida tässä kohtaa, sillä prototyypin on pitkälti tarkoitus esittää idean toimivuus. Herää silti yksi ajatus: onko hahmon muokkauksessa mitään järkeä pelissä, jota katsotaan ensimmäisestä persoonasta? Jatkokehityksen kannalta voisi olla hyvä lisätä tämä ominaisuus prototyyppiin ja tätä kautta todeta onko ominaisuudessa järkeä. Mikäli ei, on syytä pohtia muuta keinoa palkita pelaaja tai siirtyä pois ensimmäisen persoonan kuvakulmasta. Kolmas persoona voisi olla ehkä toimivampi ratkaisu, temaattisesti hahmon muokkaus käy kuitenkin järkeen.

Prototyyppiä tarkastellessa pelkästään prototyyppinä, voidaan todeta, että vaikka paljon tuotannossa ajateltiin jatkokehitystä, tämä olisi voinut olla vielä vallitsevampi ajatusmaailma. Suurin vika on siinä, että itse mekaniikkojen koodi on jäsennelty aika epämääräisesti. Joku toinen ohjelmoija ei todennäköisesti ihan heti ymmärtäisi mitä tapahtuu. Muuten oikeastaan esimerkiksi käyttöliittymäkoodi oli aika hyvin järjestettyä ja toisaalta voisi myös väittää, että prototyyppissä koodin laatu ei tavanomaisesti ole tärkeä seikka. Prototyyppi olisi voinut kaivata enemmän testausta, mutta päällisin puolin se vaikuttaa silti toimivalta ja sisältää rajauksen mukaiset asiat.

Kun prototyyppiä tarkastellaan suunnittelun kannalta, voidaan todeta, että tällä oli ehkä turhan tiukka raja. Prototyypin pohjalta ei pysty sanomaan, olisiko peli lopulta hauska. Prototyypin erittäin pelkistetty kenttäsuunnittelu ei valitettavasti ole mielenkiintoinen kilpailuympäristö. Jatkokehityksen kannalta lieneekin syytä panostaa seuraavaksi tähän.

Voidaan siis katsoa jatkokehityksen kannalta, että vaikka sekä suunnittelu ja prototyypin teko prosesseina olivat onnistuneita, prototyyppi olisi voinut antaa vähän enemmän tuloksia, mikäli sitä ei olisi rajattu niin pieneksi. Tämä olisi tietenkin tehnyt tästä työstä entistä pidemmän, joten tämän opinnäytteen kannalta voi olla parempi, että tyydyttiin vähempään. Mikäli työtä jatkaisi vielä näissä puitteissa, voisi työhön kuulua vielä jonkinlainen uusintasuunnitteluosuus, jonka yhteydessä prototyyppiä paranneltaisi vielä kertaalleen.

Vaikka työn lopputulos jäi siis vähän vajaaksi, työssä itsessään ei esiintynyt ongelmia. Työ organisoitiin kaikin osin onnistuneesti ja tästä johtuen tehdessä ei esiintynyt merkittäviä ongelmia. Lopputuloksena tehty pelisuunnitteludokumentti on myös mukavan laaja ja tehty prototyyppi antaa hyvän pohjan jatkokehitystä varten.

LÄHTEET

Dane, P. 2014. Gabe Newell hints as to why 'Half-Life 3' hasn't been revealed yet. Hakupäivä 1.6.2016, <http://gamerant.com/gabe-newell-talks-half-life-3/>.

Game Jolt. 2014. Abducted. Hakupäivä 10.6.2016, <http://gamejolt.com/games/abducted/34333>.

Freeman, T. 1997. Creating A Great Design Document. Hakupäivä 1.6.2016, http://www.gamasutra.com/view/feature/131632/creating_a_great_design_document.php.

Hartup, P. 2014. How do you make a perfect sequel to a successful game? Hakupäivä 1.6.2016, <http://www.newstatesman.com/culture/2014/05/how-do-you-make-perfect-sequel-successful-game>.

Helppi, V-V. 2013. Best Practice #2: Test early, test often - testing as part of your app development. Hakupäivä 14.6.2016, <http://testdroid.com/tech/test-early-test-often-testing-as-part-of-your-app-development>.

Icsharpcode. 2016. ILSpy. Hakupäivä 10.6.2016, <http://ilspy.net/>.

Info Entrepreneurs. 2009. Understand your competition. Hakupäivä 14.6.2016, <http://www.infoentrepreneurs.org/en/guides/understand-your-competitors/>.

Leonov, I. 2014. Tips to developers: how to make a good sequel to your game. Hakupäivä 1.6.2016, http://gamasutra.com/blogs/levgenLeonov/20141201/231105/Tips_for_Developers_How_to_Make_a_Good_Sequel_to_Your_Game.php.

Metro News. 2015. Who cares about video game sequels? Hakupäivä 1.6.2016, <http://metro.co.uk/2015/11/01/who-cares-about-video-game-sequels-readers-feature-5472502/>.

Outsoft. 2015. A better game development strategy: sequels. Hakupäivä 1.6.2016, <http://outsoft.com/blog/a-better-game-development-strategy-sequels>.

Sayenko, A. 2015. How (and Why) to Write a Great Game Design Document. Hakupäivä 1.6.2016, <http://gamedevelopment.tutsplus.com/articles/how-and-why-to-write-a-great-game-design-document--cms-23545>.

Jenkins, D. 2012. Dark Souls interview – hard master. Hakupäivä 14.6.2016, <http://metro.co.uk/2012/08/29/dark-souls-interview-hard-master-556118/>.

StandardGamer. 2014. What makes a good video game sequel? Hakupäivä 1.6.2016, <http://www.ign.com/blogs/standardgamer/2014/11/27/what-makes-a-good-video-game-sequel>.

Sulopuisto, O. 2012. EU-tuomioistuin: takaisinmallinnus ei riko tekijänoikeutta. Hakupäivä 10.6.2016, <http://www.tivi.fi/Arkisto/2012-05-14/EU-tuomioistuin-takaisinmallinnus-ei-riko-tekij%C3%A4noikeutta-3192082.html>.

Tech-Next. 2010. Unity3d decompiler. Hakupäivä 10.6.2016, <http://en.unity3d.netobf.com>.

Unity Technologies. 2016. Optimizing scripts. Hakupäivä 10.6.2016, <http://docs.unity3d.com/Manual/MobileOptimizationPracticalScriptingOptimizations.html>.

Zucconi, A. 2015. A practical tutorial to hack (and protect) Unity games. Hakupäivä 10.6.2016, <http://www.alanzucconi.com/2015/09/02/a-practical-tutorial-to-hack-and-protect-unity-games/>.